UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

SINGULAR COMPUTING LLC,   )

)   Civil Action No. 1:24-cv-10008

)

      Plaintiff,   )

)

v.   )

)

GOOGLE LLC,   )

)

      Defendant.   )   **JURY TRIAL DEMANDED**

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff, Singular Computing LLC ("Singular"), for its complaint against Defendant,

Google LLC, ("Google"), alleges as follows:

THE PARTIES

1.   Singular is a Delaware limited liability company having its principal places of

business at 10 Regent Street, Newton, MA 02465 and The Cambridge Innovation Center, 1

Broadway, Cambridge, MA 02142.

2.   Google is a Delaware limited liability company and has regular and established

places of business in this District, including a major office complex in Cambridge,

Massachusetts with over 1,500 employees. Google may be served with process through its

registered agent, Corporation Service Company, 84 State Street, Boston, MA 02109.

1

JURISDICTION

3.   This is a civil action for patent infringement under the patent laws of the United

States, 35 U.S.C. § 271, et seq. This Court has subject matter jurisdiction under 28 U.S.C.

§§ 1331 and 1338(a).

4.   This Court has general personal jurisdiction over Google because Google is

engaged in substantial activity, which is not isolated, at its regular and established places of

business within this judicial district. This Court has specific jurisdiction over Google because

Google has committed acts of infringement within this judicial district giving rise to this action,

and has established more than minimum contacts within this judicial district, such that the

exercise of jurisdiction over Google in this Court would not offend traditional notions of fair

play and substantial justice.

5.   Venue is proper in this judicial district pursuant to 28 U.S.C. §§ 1391(b)-(c) and

1400(b) because Google maintains regular and established places of business and has committed

acts of patent infringement within this judicial district.

FACTUAL BACKGROUND

6.   Singular was founded by Dr. Joseph Bates to, *inter alia*, design, develop, and

produce computers having new architectures, including the patented computer architectures at

issue in this case. Dr. Bates is the President and CEO of Singular. Since 2009, Singular has

continuously operated out of the Boston area.

7.   Dr. Bates' interest in computer science dates back to at least 1969, when, at the

age of thirteen, he was admitted to Johns Hopkins University as an undergraduate. His success

at this university sparked a pilot program for exceptionally gifted youths. This program went on

to become the widely recognized Johns Hopkins Center for Talented Youth (also known as

"CTY"; *see* https://cty.jhu.edu) program that developed the talents of over 165,000 academically advanced pre-college students, including Google's co-founder (Sergei Brin). By age seventeen, Dr. Bates had earned bachelor's and master's degrees in computer science from Johns Hopkins. He then earned a computer science doctoral degree from Cornell University at age twenty-three. Dr. Bates' research and teaching interests have centered around several cutting-edge computer science topics, including formal logic, the design and implementation of computer programming languages, and artificial intelligence ("AI").

8.   During his career working at the vanguard of computer science, Dr. Bates realized that although the theoretical computing power inside computers (as represented by the number of transistors inside a computer) was growing exponentially under a phenomenon popularly known as Moore's Law, the vast majority of that increase in computing power was not being made available to users. Under then existing computer architectures, even computers containing over a billion transistors were architected to typically perform only a handful of operations per unit of time ("clock cycle," "cycle," or "period") when using central processing units ("CPUs"). Such conventional computers typically performed only a few hundred operations per cycle when using graphics processing units ("GPUs").

9.   In the course of his work, Dr. Bates realized that existing computing architectures prevented computers from achieving their full potential. Computers perform computations using transistors (semiconductor devices that control the flow of electric current). For the last 50 years, due to advances in semiconductor technology, the number of transistors inside computer chips has grown at an exponential rate, doubling roughly every two years. Computer chips in the early 1970s contained just a few thousands transistors, while many similar chips used today have over 10 *billion* transistors. Dr. Bates recognized, however, that computing power (as measured by,

3

e.g., the number of computations a computer performs each second) had not increased at the same rate. Dr. Bates further recognized that computing power gains were lagging behind transistor count gains because a computer built using a conventional architecture – even though it included more transistors – did not use transistors efficiently.

10.   Dr. Bates devised improved computer architectures that allow a computer to make more efficient use of its physical resources (*e.g.*, its transistors). The novel architectures invented by Dr. Bates involve computer processors that contain a large number of processing elements designed to perform low precision high dynamic range ("LPHDR") operations. In these novel architectures, numerical values can be represented and manipulated inside processing elements using smaller bit widths (at the cost of lower precision), which in turn enables such processing elements to be smaller than processing elements that perform corresponding traditional high-precision operations. The relatively small size of LPHDR processing elements enables a greater number of them to be packed inside a computer chip, and operated in parallel with each other, which increases the number of operations performed per clock cycle by that chip by sacrificing the accuracy to a certain but tolerable extent. These architectures thus allow computers to use a given number of transistors more efficiently, while maintaining a high dynamic numerical range so as to have broad applicability to a wide variety of computer software programs. Dr. Bates' inventive computer architecture is broadly applicable to AI software programs, for example. In fact, Dr. Bates' inventive computer architecture has revolutionized the field of AI by vastly increasing the speed and performance of computer processors when executing AI software programs.

11.   A key difference between conventional computer architectures and Dr. Bates' invention relates to a computer's performance of arithmetic operations (as one example,

multiplication). Using a conventional computer architecture, a typical multiplier circuit inside a conventional processing element contains on the order of a hundred thousand transistors or more. A computer built using Dr. Bates' patented architecture, on the other hand, includes low-precision multiplier circuits inside processing elements that require a far smaller number of transistors, making it possible to include a very large number of them on a single chip, thereby increasing the number of multiplications per clock cycle the computer is able to perform. Indeed, a computer architected based on Dr. Bates' invention can potentially perform hundreds of times more multiplications per clock cycle, and therefore hundreds of times more multiplications per second, than a conventional computer having the same number of transistors.

12.   In some embodiments of Singular's patented computer architectures, processing elements that operate at low precision can be deployed within a computer in parallel configurations to further amplify their relatively higher efficiency. In still other exemplary configurations, large numbers of these LPHDR processing elements can be deployed in conjunction with far smaller numbers of traditional high-precision processing elements found within conventional computer architectures.

13.   Singular's revolutionary approach to computer architecture is described in a provisional patent application entitled "Massively Parallel Processing with Compact Arithmetic Element" that was filed in June of 2009 and made public in June of 2010, and in a non-provisional application entitled "Processing with Compact Arithmetic Processing Element" that was filed in June of 2010 and made public in April of 2012.

14.   After Dr. Bates filed these patent applications, he built a prototype computer using his novel architecture. The Singular prototype was able to execute a software program that, for example, was able to perform neural network image classification (an AI application) thirty

times faster than a conventional computer having comparable physical characteristics in terms of its number of transistors, its semiconductor fabrication process and its power draw.

15.   As Singular was designing and building prototypes of its new computer, Google was belatedly recognizing the limitations of its conventional computer architectures in providing users with computer-based services such as Translate, Photos, Search, Assistant, and Gmail. According to Google, these limitations caused a "scary and daunting" situation for Google. The situation arose as Google was starting to improve these computer-based services by basing them on AI software programs that were running on its conventional computers. The situation was "scary and daunting" because the new AI software programs required far more computer operations per period of time (i.e., an hour, a day, etc.) than its conventional computers could provide. For example, by Google's own estimation, applying its new AI software programs to speech recognition services alone (e.g., Translate and Assistant) would increase the required number of operations per period of time so drastically that Google would have to at least double its total computing footprint.

16.   As Google's scary and daunting situation unfolded, representatives from Google formally met with Dr. Bates to discuss his invention at least five times in person and at least 15 times by telephone, from 2010 through early 2017. During the course of these meetings and calls, Dr. Bates disclosed his computer architectures and prototype. Dr. Bates also advised Google such was patent-protected. On March 1, 2011, less than 2 years after the filing of the provisional application, Dr. Bates and Google executed a non-disclosure agreement ("NDA") prepared by Google with an effective date of November 1, 2010. After years of conversations over email and phone, as well as multiple in-person meetings, Google proposed another NDA to

Bates, in March of 2017, this one asking Dr. Bates to waive any potential willful patent infringement claims against Google.

17.   Following disclosure to Google by Dr. Bates of his invention between 2010 and 2017, Google copied and adopted Dr. Bates' patented invention, incorporating such initially in its TPUv2 and TPUv3 chips (the accused products in Case No. C.A. No. 1:19-cv-12551-FDS), and later in its TPUv4 and TPUv5 chips, which share several features with the TPUv2 and TPUv3 chips (https://cloud.google.com/tpu/docs/system-architecture-tpu-vm) (the TPUv4 and TPUv5 chips individually and/or together, "Accused TPU Devices") and in instances of the Cloud TPU computing system that includes the Accused TPU Devices ("Accused TPU Computing System"). This is apparent from a comparison of Dr. Bates' patented architecture and that of the Accused TPU Devices and Accused TPU Computing Systems. It is also apparent from an exemplary comparison of the disclosures made in writing by Dr. Bates to Google from 2010-2017 on one hand, with the properties and features that Google later adopted in its Accused TPU Devices and Accused TPU Computing System on the other hand. For example:

Singular Presentations Made to Google / Jeff Dean (2010-2014)	**Google Documents**
	Google Publication of TPU

(SINGULAR'S)
APPROXIMATE COMPUTING

A traditional massively parallel machine, with floating point arithmetic that is "99% correct"

(e.g. $1.0 + 1.0 = 1.98 \ldots 2.02$)

- **Surprise #1:** Arithmetic circuit can be unexpectedly small

Standard FPU ~500,000 transistors

~100x smaller standard deterministic digital cmos

+ - * / sqrt one cycle per op

Special computation properties

reduced precision ok

| about 1.2 |
| × about 0.6 | **NOT** | 1.2104 × 0.61127 |
| about 0.7 | | 0.73989143 |

handful of specific operations

OTHER PROMISING DOMAINS
(IN PROGRESS - INITIAL EVIDENCE)

- Vision: segmenting smooth objects (weak features, Hartmut/Joe intuition)

- Molecular dynamics, Protein folding (all-atom energy)

- Genomics (eg Smith-Waterman dynamic programming)

- Machine learning (neural nets, genetic algorithms with local crossover, local graphical models, simulated annealing)

- Speech recognition (HMMs, many concurrent voice streams, Dragon CTO)

- Neocortex sim (>human, faster than realtime, supercomputer $)

Confidential Property of Singular Computing   June 2011   19

Machine Learning for Systems and Systems for Machine Learning

Jeff Dean
Google Brain team
g.co/brain

Presenting the work of **many** people at Google

Combine FPU with 200 words memory. Build 2D grid,

Host CPU

PE mem+fpu | PE

Control Unit

PE | PE

Secondary Storage (DRAM)

In vo

4K

1M
(

Confidential Property of Singular Computing   June 2011

How TPU works   Google Cloud

MULTIPLY & ADD

8   9

HBM 8 GB | core, scalar/vector units | core, scalar/vector units | HBM 8 GB

MXU 128x128 | MXU 128x128

Jeff Dean, *Machine Learning for Systems and Systems for Machine Learning*, dean-nips17.pdf (learningsys.org).

18.   Google knew that its demand for AI-based user services far exceeded its computing capabilities. Google recognized that, but for its inclusion of the technology covered by Dr. Bates' patents inside its computers, it would have had to at least double its computing footprint to accommodate such demand for delivering increased speech recognition services alone.

19.   Google realized it needed to use Dr. Bates' computer architectures to increase the number of computer operations per period of time that could be executed by its computers. To this end, it has been using the Accused TPU Devices and Accused TPU Computing Systems to deliver services such as Translate, Photos, Search, Assistant, Cloud, and Gmail to the public. Google drives the public's use of these services to enhance its Ads platform which, in turn, generates at least tens of billions of dollars per year in profit for Google. *See* https://www.statista.com/statistics/266206/googles-annual-global-revenue/.

20.   Google now operates at least fourteen data centers in the United States for its TPU computers. *See* www.google.com/about/datacenters/locations/. The Accused TPU Devices and Accused TPU Computing Systems are installed and operated by Google in one or more of Google's data centers at: Berkeley County, South Carolina; Council Bluffs, Iowa; The Dalles, Oregon; Douglas County, Georgia; Henderson, Nevada; Jackson County, Alabama; Lenior, North Carolina; Loudoun County, Virginia; Mayes County, Oklahoma; Midlothian, Texas; and Montgomery County, Tennessee; New Albany, Ohio; Papillion, Nebraska; and Storey County, Nevada.

21.   In the Securities and Exchange Commission Form 10-K filed by Google's parent Alphabet, Inc. ("Alphabet") for the fiscal year ending December 31, 2022, Alphabet reported net

income of approximately $74.8 billion for 2022 revenues in excess of $282 billion. *See*

https://abc.xyz/assets/investor/static/pdf/20230203_alphabet_10K.pdf?cache=5ae4398.

<div align="center"><u>THE PATENTS-IN-SUIT</u></div>

22.   On May 10, 2022, the United States Patent and Trademark Office ("USPTO")

issued United States Patent No. 11,327,714 ("the '714 patent"), titled PROCESSING WITH

COMPACT ARITHMETIC PROCESSING ELEMENT. On August 25, 2020, the USPTO

issued United States Patent No. 10,754,616 ("the '616 patent"), titled PROCESSING WITH

COMPACT ARITHMETIC PROCESSING ELEMENT. On November 9, 2021, the USPTO

issued United States Patent No. 11,169,775 ("the '775 patent"), titled PROCESSING WITH

COMPACT ARITHMETIC PROCESSING ELEMENT. On September 26, 2023, the USPTO

issued United States Patent No. 11,768,659 ("the '659 patent"), titled PROCESSING WITH

COMPACT ARITHMETIC PROCESSIGN ELEMENT. On September 26, 2023, the USPTO

issued United States Patent No. 11,768,660 ("the '660 patent"), titled PROCESSING ELEENT

WITH COMPACT ARITHMETIC PROCESSING ELEMENT. On December 12, 2023, the

USPTO issued United States Patent No. 11,842,166 ("the '166 patent"), titled PROCESSING

ELEENT WITH COMPACT ARITHMETIC PROCESSING ELEMENT. The '714 patent, '616

patent, '775 patent, '659 patent, '660 patent and '166 patent, (collectively the "patents-in suit" or

"Asserted Patents"), are each valid and enforceable.

23.   The application to which the patents-in-suit claim priority (No. 61/218,691) was

filed on June 19, 2009.

24.   Singular is the owner and assignee of all rights, title, and interest in and to the

patents-in-suit, and holds all substantial rights therein, including the right to grant licenses, to

exclude others, and to enforce and recover past damages for infringement.

25.   Google's infringement of the '714 patent, '616 patent, '775 patent, '659 patent,

'660 patent and '166 patent is willful.

26.   Google knew or should have known of at least the '616 patent and the '775

patent. In December 22, 2021, Singular sued Google for infringement of these patents in this

District (case 1:21—cv-12110). The parties agreed to a dismissal of that case without prejudice

on April 20, 2023.

PATENT ELIGIBILITY - CLAIMED ADVANCE OF REPRESENTATIVE CLAIMS

27.   The claims asserted in this action are eligible for patenting under 35 U.S.C. § 101.

28.   Claim 1 of the '714 patent recites the following limitations, each of which is

found in the Accused TPU Devices as set forth below:

A device comprising:

at least one instruction memory adapted to store at least one instruction;

a silicon chip comprising a plurality of first execution units, wherein each of the plurality of first execution units has access to memory local to that execution unit and is adapted to execute a first operation of multiplication: on one or more first input signals that represent a first numerical value using a floating-point representation, and on one or more second input signals that represent a second numerical value using a floating-point representation, to produce one or more first output signals that represent a third numerical value, wherein the dynamic range of possible valid inputs to the first operation is at least as wide as from 1/1,000,000,000 through 1,000,000,000 and for each of at least X=10% of the possible valid inputs to the first operation the numerical value represented by the one or more first output signals differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that input;

a second execution unit adapted to execute a second operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide; and

decoding circuitry adapted to decode the at least one instruction received from the at least one instruction memory and to send at least one control signal to at least one of the plurality of first execution units to cause the at least one of the plurality of first execution units to operate according to the at least one instruction;

wherein a total number of first execution units in the silicon chip exceeds, by at least 100 more than five times, a total number of execution units in the silicon chip adapted to execute the operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide;

wherein each of the plurality of first execution units is smaller than the second execution unit; and

wherein the plurality of first execution units are adapted to collectively perform, per cycle, at least tens of thousands of the first operation.

29.   Claim 7 of the '616 patent recites the following limitations, each of which is found in the Accused TPU Computing System as set forth below:

A computing system, comprising:

a host computer;

a computing chip comprising:

> *a processing element array comprising a plurality of first processing elements, wherein the plurality of first processing elements is no less than 5000 in number, wherein each of a first subset of the plurality of first processing elements is positioned at a first edge of the processing element array, and wherein each of a second subset of the plurality of first processing elements is positioned in the interior of the processing element array;*
> *an input-output unit connected to each of the first subset of the plurality of first processing elements;*
>
> *a plurality of processing element connections, each processing element connection connecting one of the plurality of first processing elements with another of the plurality of first processing elements, wherein each of the plurality of first processing elements is connected to at least one other of the plurality of first processing elements by at least one of the plurality of processing element connections;*
>
> *a plurality of memory units, wherein each of the plurality of first processing elements is associated with a corresponding one of the plurality of memory units, and wherein each of the plurality of memory units is local to its associated one of the plurality of first processing elements; and,*

12

a plurality of arithmetic units, wherein each of the plurality of first processing elements has positioned therein at least one of the plurality of arithmetic units; and,

a host connection at least partially connecting the input-output unit with the host computer;

wherein the plurality of arithmetic units each comprises a first corresponding multiplier circuit adapted to receive as a first input to the first corresponding multiplier circuit a first floating point value having a first binary mantissa of width no more than 11 bits and a first binary exponent of width at least 6 bits, and to receive as a second input to the first corresponding multiplier circuit a second floating point value having a second binary mantissa of width no more than 11 bits and a second binary exponent of width at least 6 bits.

30.   Claim 7 of the '775 patent recites the following limitations, each of which is found in the Accused TPU Computing Systems as set forth below:

A computing system, comprising:

a host computer;

a computing chip comprising:

> *a processing element array comprising a plurality of first processing elements, wherein the plurality of first processing elements is no less than 5000 in number, wherein each of a first subset of the plurality of first processing elements is positioned at a first edge of the processing element array, and wherein each of a second subset of the plurality of first processing elements is positioned in the interior of the processing element array;*
>
> *an input-output unit connected to each of the first subset of the plurality of first processing elements;*
>
> *a plurality of processing element connections, each processing element connection connecting one of the plurality of first processing elements with another of the plurality of first processing elements, wherein each of the plurality of first processing elements is connected to at least one other of the plurality of first processing elements by at least one of the plurality of processing element connections;*
>
> *a plurality of memory units, wherein each of the plurality of first processing elements is associated with a corresponding one of the plurality of memory units, and wherein each of the plurality of memory units is local to its associated one of the plurality of first processing elements;*

13

a plurality of first arithmetic units, wherein each of the plurality of first processing elements has positioned therein at least one of the plurality of first arithmetic units;

a plurality of second processing elements; and

a plurality of second arithmetic units, wherein each of the plurality of second processing elements has positioned therein at least one of the plurality of second arithmetic units; and

a host connection at least partially connecting the input-output unit with the host computer;

wherein the plurality of first arithmetic units each comprises a first corresponding multiplier circuit adapted to receive as a first input to the first corresponding multiplier circuit a first floating point value having a first binary mantissa of width no more than 11 bits and a first binary exponent of width at least 6 bits, and to receive as a second input to the first corresponding multiplier circuit a second floating point value having a second binary mantissa of width no more than 11 bits and a second binary exponent of width at least 6 bits;

wherein the first multiplier circuits corresponding to the plurality of first arithmetic units each comprises a first respective plurality of transistors and has no other transistors;

wherein the plurality of second arithmetic units each comprises a second corresponding multiplier circuit adapted to receive as inputs to the second corresponding multiplier circuit two floating point values each of width at least 32 bits;

wherein the second multiplier circuits corresponding to the plurality of second arithmetic units each comprises a second respective plurality of transistors;

wherein each of the second respective pluralities of transistors of the second multiplier circuits corresponding to the plurality of second arithmetic units exceeds in number each of the first respective pluralities of transistors of the first multiplier circuits corresponding to the plurality of first arithmetic units.

31.   Claim 1 of the '659 patent recites the following limitations, each of which is

performed in the Accused TPU Devices as set forth below:

A method, for use with a silicon chip that has a clock, the method comprising:

completing, in a single cycle of the clock, using the silicon chip, at least tens of thousands of first multiplication operations;

wherein each of the first multiplication operations operates on

a respective first numerical input value represented using a first floating point representation that has a signed binary mantissa of no more than 11 bits and a signed binary exponent of at least 6 bits,

and a respective second numerical input value represented using a second floating point representation; and

wherein the number of the first multiplication operations is at least 1000 more than three times the maximum number of traditional high-precision multiplication operations on floating point numbers at least 32 bits wide that the silicon chip is adapted to complete in a single cycle of the clock.

32.   Claim 1 of the '660 patent recites the following limitations, each of which is

found in the Accused TPU Devices as set forth below:

A device comprising:

a silicon chip comprising a plurality of execution units;

wherein the plurality of execution units jointly comprise a first plurality of custom silicon arithmetic elements;

wherein at least one of the first plurality of custom silicon arithmetic elements is adapted to execute a first multiplication operation

on one or more first input signals that represent a first numerical value using a floating point representation that has a signed binary mantissa of no more than 11 bits and a signed binary exponent of at least 6 bits,

and on one or more second input signals that represent a second numerical value using a floating point representation;

wherein a total number of the first plurality of custom silicon arithmetic elements in the silicon chip that are adapted to execute first multiplication operations exceeds, by at least 1000 more than three times, a total number of second custom silicon arithmetic elements in the silicon chip adapted to perform on each cycle the operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide; and

wherein the first plurality of custom silicon arithmetic elements are adapted to collectively perform, per cycle, at least tens of thousands of first multiplication operations.

15

33.   Claim 1 of the '166 patent recites the following limitations, each of which is found in the Accused TPU Devices as set forth below:

A device comprising:

at least one instruction memory adapted to store at least one instruction;

a silicon chip comprising a plurality of first execution units, wherein each of the plurality of first execution units has access to memory local to that execution unit and is adapted to execute a first operation of multiplication:

on one or more first input signals that represent a first numerical value using a floating-point representation that has a signed binary mantissa of no more than 11 bits and a signed binary exponent of at least 6 bits, and on one or more second input signals that represent a second numerical value using a floating-point representation, to produce one or more first output signals that represent a third numerical value;

a second execution unit adapted to execute a second operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide; and

decoding circuitry adapted to decode the at least one instruction received from the at least one instruction memory and to send at least one control signal to at least one of the plurality of first execution units to cause the at least one of the plurality of first execution units to operate according to the at least one instruction;

wherein a total number of first execution units in the silicon chip exceeds, by at least 100 more than five times, a total number of execution units in the silicon chip adapted to execute the operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide;

wherein each of the plurality of first execution units is smaller than the second execution unit; and

wherein the plurality of first execution units are adapted to collectively perform, per cycle, at least tens of thousands of the first operation.

34.   The execution units or processing elements of the foregoing claims are each a circuit that is part of a silicon chip or computing chip, which chips are themselves part of a device (i.e., a computer) or computing system. As instructed by a computer program, these "execution units" each performs operations on an input signal representing a first numerical value and on another input signal representing a second numerical value, to produce a first output signal representing a third numerical value.

16

35.   In the foregoing claims, all operations are performed on input signals representing

a first numerical value having a "high dynamic range" at least as wide as from 1/1,000,000,000

through 1,000,000,000. The "dynamic range" is the range of the value of inputs that can be

operated upon. The dynamic range is typically expressed as being at least as wide as "from 1/Z

through Z." Floating point representations of input values (i.e., floating point numbers) that

include a signed binary exponent field of at least 6 bits, have a dynamic range of at least

1/1,000,000,000 through 1,000,000,000 (i.e., an exponent field of 6 bits can have a maximum

value of 64 since 2^6=64, and accounting for the bias that would be applied to the actual exponent

value in popular floating point formats, the dynamic range for a 6 bit exponent field is from 2^{31}

(which is > 1,000,000,000) to 2^{-30} (which is $< \frac{1}{1,000,000,000}$)). Multiplication operations on inputs

having this dynamic range are "high dynamic range" operations, and execution units or

processing elements that execute such "high dynamic range" operations are "high dynamic

range" execution units or processing elements respectively.

36.   In some of the foregoing claims, the operation of multiplication produces an

output signal that represents an output numerical value that for at least X = 10% of the possible

valid inputs to the operation, differs by at least 0.05% from the result of an exact mathematical

operation on the numerical values of the same input. Under some of the foregoing claims, if a

multiplication operation is performed on two input signals that are each floating point

representations of a value, the operation generates an output signal that represents an output

numerical value, which for at least X = 10% of the possible valid inputs to the operation differs

by at least 0.05% from the result of an exact mathematical operation on the numerical values of

the same inputs. Multiplication operations having this error distribution are "low precision"

operations, and execution units or processing elements that execute such "low precision"

operations are "low precision" execution units or processing elements respectively.

37.   In some of the foregoing claims, the operation of multiplication is performed on

two input signals each representing a first numerical value using a floating point representation

that includes a signed binary exponent field of at least 6 bits and a signed binary mantissa field of

no more than 11 bits. The representation error of such numerical values, when representing a

value B for example, can be as high as 0.1% compared to a traditional single-precision

representation of the value B. Multiplication operations on such numerical values are also "low

precision" operations herein, and execution units or processing elements that operate on such

numerical value are also "low precision" execution units or processing elements respectively.

38.   Multiplication operations having the "high dynamic range" described in

paragraph 35 and the "low precision" described in either paragraph 36 or paragraph 37 are "low

precision high dynamic range" or "LPHDR" operations. Execution units (or processing

elements) having the "high dynamic range" described in paragraph 35 and the "low precision"

described in either paragraph 36 or paragraph 37 are "low precision high dynamic range" or

"LPHDR" execution units (or processing elements).

39.   In some of the foregoing claims, a "second execution unit" is configured to

execute a second operation of traditional high-precision multiplication. This operation is

performed on the Institute of Electrical and Electronics Engineers (IEEE) single-precision or

double-precision floating point numbers that are at least 32 bits wide, which have high dynamic

range. In a conventional multiplication operation performed on two input signals that are each

single-precision floating point representations of a value, the operation does not generate an

output signal that represents an output numerical value with low precision. Such multiplication

operations are "traditional high-precision" operations, and the corresponding execution units (or processing elements) are "traditional high-precision" execution units (or processing elements), and not LPHDR operations, execution units, or processing elements respectively.

40.   An example of the invention of the foregoing claims is a computer having a first number of first execution units, where each first execution unit is structured to receive as inputs electrical signals that represent numbers using a logarithmic number system format with a signed integer field of 6 bits and a signed fraction field of 7 bits. *See* Figure 5 of the '714 patent below.
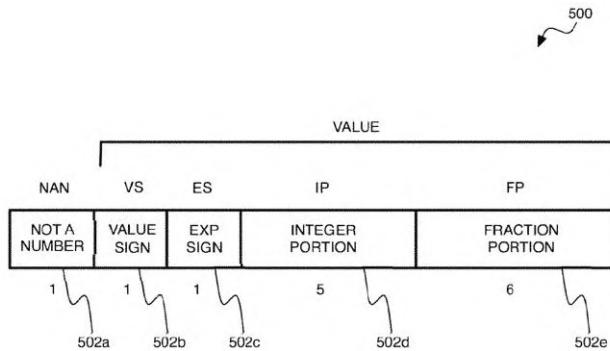

FIG. 5

Each such execution unit is an LPHDR execution unit. This example computer further includes a second number of traditional high-precision execution units that is far smaller than the first number of LPHDR execution units.

41.   Another example of the invention of claim 1 is a computer having a first number of first execution units, where each first execution unit is structured to receive as inputs electrical signals that represent numbers using a floating point number system format with a signed exponent field of at least 6 bits and a signed mantissa field of no more than 11 bits. This would include a computer having a first number of first execution units, where each first execution unit is structured to receive as inputs electrical signals that represent numbers using the "brain float

16" (bfloat16) number format, which is a floating point number system format with a signed

exponent field of 8 bits and a signed mantissa field of 8 bits.
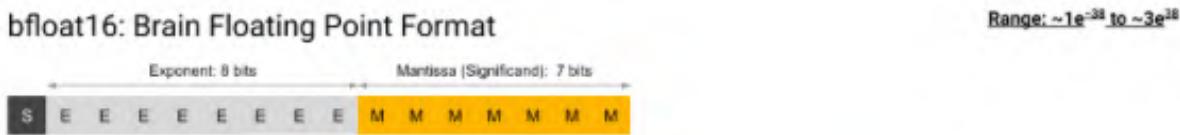
bfloat16: Brain Floating Point Format   Range: ~1e⁻³⁸ to ~3e³⁸

Each such execution unit is an LPHDR execution unit. This example computer further includes a

second number of traditional high-precision execution units that is far smaller than the first

number of LPHDR execution units.

42.   The devices, computer systems and chips of the foregoing claims, and the

devices, computer systems and chips that perform the methods of the foregoing claims,

substantially differ structurally from devices, computer systems and chips in the prior art. For

example, graphics processors that included support for traditional IEEE half precision 16 bit

floating point, alongside support for 32 bit floating point and 64 bit floating point, as disclosed in

the each of the Asserted Patents, did not have any execution units or processing elements that

receive as an input an electrical signal representing numbers having a dynamic range at least as

wide as from 1/1,000,000 through 1,000,000, and that transmit as an output for at least $X = 10\%$

of the possible valid inputs to that operation, an electrical signal representing numbers that differ

by at least 0.05% from the result of an exact mathematical calculation of that operation on the

numerical values of that same input. Such graphics processors also did not have execution units

or processing elements structured to receive as inputs electrical signals that represent numbers

using a floating point number system format with a signed exponent field of at least 6 bits and a

signed mantissa field of no more than 11 bits. The graphics processors disclosed in the Asserted

Patents did not include a single such low precision high dynamic range execution unit or

processing element. Dr. Bates was the first to reduce to practice a computer in which *a total*

*number of such LPHDR execution units (*or processing elements) *in a silicon chip is no less than 5000,* or *exceeds by a very large number (at least 100 more than five times) a total number of execution units (*or processing elements) *in the silicon chip adapted to execute the operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide.* Dr. Bates even coined "LPHDR" to describe such novel low precision high dynamic range execution units. As computers that included such numbers of LPHDR execution units or processing elements did not exist in the prior art, Dr. Bates' invention is not the use of existing computing technology or an existing implementation of a computer processor.

43.   The devices, computer systems and chips of the foregoing claims, and the devices, computer systems and chips that perform the methods of the foregoing claims, have many advantages over the devices, computer systems and chips using conventional architectures. They include, but are not limited to, the following advantages:

a.   including many more multiplier circuits on a single computer chip having a given set of resources, such as transistors, than prior art computer chips having a similar set of resources, by utilizing relatively imprecise multiplication circuits that represent and manipulate numerical values using smaller bit widths and thus requiring far fewer transistors than conventional, full-precision multiplication circuits;

b.   performing a far greater number of operations per clock cycle – potentially on the order of 100 times or more – than a conventional computer of the time having the same number of transistors, semiconductor fabrication process and power draw; and

c.   supporting software programs that require operations to be performed on numbers having high dynamic range.

44.   The devices, computer systems and chips of the foregoing claims, and the

devices, computer systems and chips that perform the methods of the foregoing claims, are able

to execute a far larger number of multiplication operations (for example) per period of time than

their conventional counterparts, while supporting software programs that require multiplication

operations to be performed on numbers having a high dynamic range. By deploying massive

numbers of LPHDR execution units or processing elements in conjunction with far smaller

numbers of traditional high-precision execution units or processing elements, the device supports

operations performed at a wider range of precisions and dynamic ranges while still performing at

previously unseen levels of computing efficiency. Furthermore, by incorporating a computing

device having such ratios of low and high precision execution units, these novel devices,

computer systems and chips each implement a heterogeneous architecture that can support a

wider range of software programs.

45.   The computer systems of some of the foregoing claims comprise a new

computing architecture further comprising a massively parallel array of LPHDR execution units

or processing elements ("processing element array" or "PEA") and a host responsible for overall

control. The host seeks to have the PEA perform massive amounts of computations in a useful

way by causing the PEA's execution units or processing elements to each perform computations

typically on data locally stored in that execution unit, with all the execution units or processing

elements performing those computations in parallel with one another. To most efficiently control

the PEA, the system may include a specialized control unit (CU) having the primary task of

retrieving and decoding instructions from an instruction memory and issuing partially decoded

instructions to all the execution units or processing elements in the PEA. The computer systems

of these claims enable hosts, which may themselves be conventional, to control a computer

system that performs massive amounts of arithmetic using a small amount of computing

resources (e.g., transistors or volume) relative to conventional computer systems which could not

perform such amounts of arithmetic using such a small amount of computing resources.

46.   The claimed advances achieved by the PEA of some of the foregoing claims—

primarily the ability to perform massive amounts of arithmetic using a small amount of

computing resources (e.g., transistors or volume) relative to conventional computer systems—is

coupled with another claimed advance achieved by Dr. Bates' claimed novel computer

architectures. Some of the foregoing claims further comprise an input/output unit (IOU) to serve

as an interface between the host and various peripherals in the computing system on the one

hand, with the PEA on the other hand. More specifically, in some of the foregoing claims, the

I/O unit is connected to execution units or processing elements positioned at one of more edges

of the PEA, and those edge execution units or processing elements in turn are connected to

execution units or processing elements positioned inside (i.e, not at the edges) of the PEA. This

arrangement of I/O units and the PEA, enables the PEA to operate at or near its peak

computational throughput, even though the host performs operations far less quickly than the

PEA, by reducing the number of PEAs (and therefore the amount of data moved into and outout

of the PEA per period of time) with which the host has to interface directly. Put another way, this

arrangement prevents the computer architecture created by Dr. Bates, with its PEA that operates

so much faster than the host, from becoming I/O bound (i.e., because the PEA processes data far

faster than any conventional host that is part of the same computing system). Importantly, this

claimed arrangement can achieve this crucial advantage while minimizing the number of long

distance transfers within the computer system (e.g., such long distance transfers would arise for

example if the I/O unit had to interact with processing elements at the far end of a PEA), and without requiring large amounts of hardware resources.

COUNT I
(Google's Infringement of United States Patent No. 11,327,714)

47.   Paragraphs [1-46] are reincorporated by reference as if fully set forth herein.

48.   Google has directly infringed, and continues to directly infringe, literally and/or by the doctrine of equivalents, at least claim 1 of the '714 patent by making, using, testing, selling, offering for sale and/or importing into the United States a plurality of Accused TPU Devices that are grouped within a pod ("Accused TPU Pod"). The Accused TPU Devices, in Google's own words, "power" at least Google Translate, Photos, Search, Assistant, and Gmail, as published by Google:

Empowering businesses with Google Cloud AI

Machine learning has produced business and research breakthroughs ranging from network security to medical diagnoses. We built the Tensor Processing Unit (TPU) in order to make it possible for anyone to achieve similar breakthroughs. Cloud TPU is the custom-designed machine learning ASIC that powers Google products like Translate, Photos, Search, Assistant, and Gmail. Here's how you can put the TPU and machine learning to work accelerating your company's success, especially at scale.

TPU Pod

A TPU Pod is a contiguous set of TPUs grouped together over a specialized network. The number of TPU chips in a TPU Pod is dependent on the TPU version.

49.   The Accused TPU Devices in the Accused TPU Pods share a similar architecture in that they all implement low precision high dynamic range arithmetic operations, specifically multiplication of two traditional high precision floating point value at reduced bfloat16 precision.

24

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The following table shows the key specifications for a v4 TPU Pod.

| Key specifications	v4 Pod values
Peak compute per chip	275 teraflops (bf16 or int8)
HBM2 capacity and bandwidth	32 GiB, 1200 GBps
Measured min/mean/max power	90/170/192 W
TPU Pod size	4096 chips
Interconnect topology	3D mesh
Peak compute per Pod	1.1 exaflops (bf16 or int8)
All-reduce bandwidth per Pod	1.1 PB/s
Bisection bandwidth per Pod	24 TB/s

The following diagram illustrates a TPU v4 chip.

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

The following diagram illustrates a TPU v5e chip.

The following table shows the key chip specifications and their values for v5e.

Key chip specifications	v5e values
Peak compute per chip (bf16)	197 TFLOPs
Peak compute per chip (Int8)	393 TFLOPs
HBM2 capacity and bandwidth	16 GB, 819 GBps
Interchip Interconnect BW	1600 Gbps

50.   An Accused TPU Pod, that groups one or more Accused TPU Devices

therewithin, is an example of a "*device*," as claimed by the '714 patent. As published by Google:

System Architecture 🔖 ▾   Send feedback

Tensor Processing Units (TPUs) are application specific integrated circuits (ASICs) designed by Google to accelerate machine learning workloads. Cloud TPU is a Google Cloud service that makes TPUs available as a scalable resource.

TPUs are designed to perform matrix operations quickly making them ideal for machine learning workloads. You can run machine learning workloads on TPUs using frameworks such as TensorFlow, Pytorch, and JAX.

TPU Pod

A TPU Pod is a contiguous set of TPUs grouped together over a specialized network. The number of TPU chips in a TPU Pod is dependent on the TPU version.

51.   Each Accused TPU Device within an Accused TPU Pod ("*device*") infringes

claim 1 of the '714 patent by *inter alia*, comprising *"at least one instruction memory adapted to*

store at least one instruction," because each Accused TPU Device contains instruction memory

IMEM.

The following diagram illustrates a TPU v4 chip.
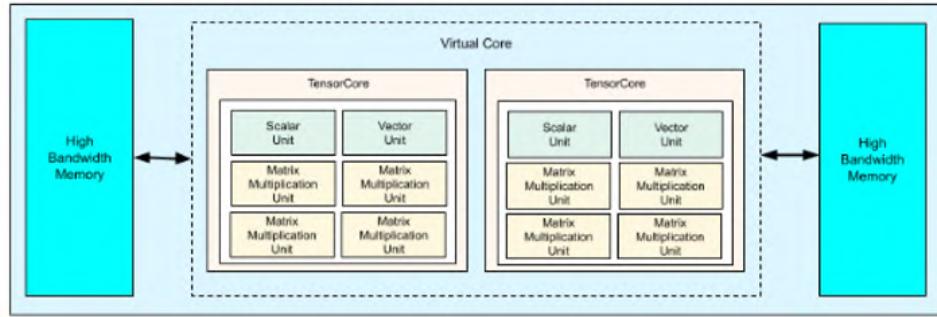
The following diagram illustrates a TPU v5e chip.
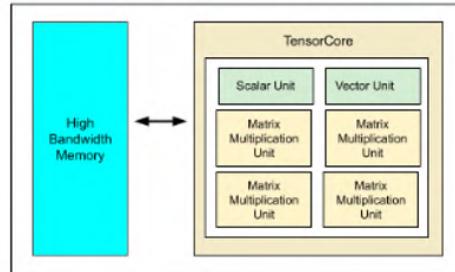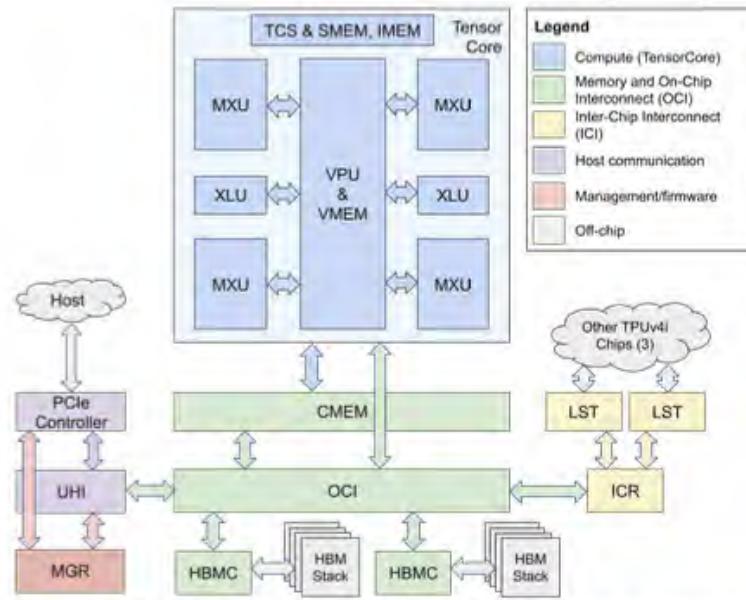
**Figure 5. TPUv4i chip block diagram. Architectural memories are HBM, Common Memory (CMEM), Vector Memory (VMEM), Scalar Memory (SMEM), and Instruction Memory (IMEM). The data path is the Matrix Multiply Unit (MXU), Vector Processing Unit (VPU), Cross-Lane Unit (XLU), and TensorCore Sequencer (TCS). The uncore (everything not in blue) includes the On-Chip Interconnect (OCI), ICI Router (ICR), ICI Link Stack (LST), HBM Controller (HBMC), Unified Host Interface (UHI), and Chip Manager (MGR).**

52.   Each Accused TPU Device infringes claim 1 of the '714 patent, by *inter alia*,

comprising "*a silicon chip comprising a plurality of first execution units, wherein each of the*

plurality of first execution units is …. adapted to execute a first operation of multiplication on

one or more first input signals that represent a first numerical value using a floating-point

representation, and on one or more second input signals that represent a second numerical value

using a floating-point representation, to produce one or more first output signals that represent

a third numerical value."

a.   Each of the Accused TPU Devices, a TPUv4 chip or a TPUv5 chip, is a silicon

chip. Each of these chips contains one or more TensorCores which each have one

of more MXUs. Specifically with respect to the Accused TPU Devices, each

TPUv4 chip has 8 MXUs (four MXUs per TensorCore, 2 TensorCores per chip),

28

and each TPUv5 chip has 4 MXUs (four MXUs per TensorCore, 1 TensorCore

per chip). As published by Google:

TPU chip

A TPU chip contains one or more TensorCores. The number of TensorCores depend on the version of the TPU chip. Each TensorCore consists of one or more matrix-multiply units (MXUs), a vector unit, and a scalar unit.

An MXU is composed of 128 x 128 multiply-accumulators in a systolic array. MXUs provide the bulk of the compute power in a TensorCore. Each MXU is capable of performing 16K multiply-accumulate operations per cycle. All multiplies take bfloat16 inputs, but all accumulations are performed in FP32 number format.

The vector unit is used for general computation such as activations and softmax. The scalar unit is used for control flow, calculating memory addresses, and other maintenance operations.

TensorCores

TPU chips have one or two TensorCores to run matrix multiplication. Similar to v2 and v3 Pods, v5e has one TensorCore per chip. By contrast, v4 Pods have 2 TensorCores per chip. For more information about TensorCores, see ACM article.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The following table shows the key specifications for a v4 TPU Pod.

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

      b.   Each MXU contains a systolic array having 128 x 128 "multiply-accumulators,"

which each include a multiplier circuit (MXU Multiplier Circuit). As published by

Google:

An MXU is composed of 128 x 128 multiply-accumulators in a systolic array. MXUs provide the bulk of the compute power in a TensorCore. Each MXU is capable of performing 16K multiply-accumulate operations per cycle. All multiplies take bfloat16 inputs, but all accumulations are performed in FP32 number format.

29

The primary task for TPUs is matrix processing, which is a combination of multiply and accumulate operations. TPUs contain thousands of multiply-accumulators that are directly connected to each other to form a large physical matrix. This is called a systolic array architecture. Cloud TPU v3, contain two systolic arrays of 128 x 128 ALUs, on a single processor.

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

c.   Each of those MXU Multiplier Circuits is associated with circuitry for taking two 32-bit floating point format ("FP32 format" or "float32") values and converting each to a bfloat16 value (an MXU Multiplier Circuit and said taking/converting circuitry, collectively an "MXU Reduced Precision Multiply Cell"). An MXU Reduced Precision Multiply Cell is a *"first execution unit."* 16,384 MXU Reduced Precision Multiply Cells are a plurality of *"first execution units."* As published by Google:

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

d.   The "*first operation of multiplication*" executed by each individual MXU

Reduced Precision Multiply Cell is a multiplication operation that is (i) performed

on two input signals each representing a float32 numerical value, but (ii) carried

out at "reduced bfloat16 precision." Such an operation (e.g., "X[2,0]*W[0,0]" in

the example equation for Y[2,0] that Google provides below) is a part of a larger

float32 matrix multiplication operation (e.g., Y= X*W in the example Google

provides below), where the individual multiplications are performed at "reduced

bfloat16 precision" by the MXU as a whole. As published by Google:

BFloat16: The secret to high performance on Cloud TPUs, Google Cloud Blog (Aug. 23, 2019), https://cloud.google.com/blog/products/ai-machine-learning/bfloat16-the-secret-to-high-performance-on-cloud-tpus.

Systolic array

The MXU implements matrix multiplications in hardware using a so-called 'systolic array' architecture in which data elements flow through an array of hardware computation units. (In medicine, 'systolic' refers to heart contractions and blood flow, here to the flow of data.)

The basic element of a matrix multiplication is a dot product between a line from one matrix and a column from the other matrix (see illustration at the top of this section). For a matrix multiplication Y=X*W, one element of the result would be:

```
Y[2,0] = X[2,0]*W[0,0] + X[2,1]*W[1,0] + X[2,2]*W[2,0] + ... + X[2,n]*W[n,0]
```


Illustration: a dense neural network layer as a matrix multiplication, with a batch of eight images processed through the neural network at once. Please run through one line x column multiplication to verify that it is indeed doing a weighted sum of all the pixels values of an image. Convolutional layers can be represented as matrix multiplications.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

e.   A respective pair of inputs to each individual MXU Reduced Precision Multiply

Cell comprise the "*first input signal*" and the "*second input signal*." The

respective pair of the first and second input signals provide to each individual

MXU Reduced Precision Multiply Cell, the respective pair of the "*first numerical*

value" and the "*second numerical value*." The "*first numerical value*" represented

by the "*first input signal*," and the "*second numerical value*" represented by the

"*second input signal*," are all float32 numbers. Float32 numbers are "*numerical*

values." As published by Google:

> Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit
> (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU
> chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total
> system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts
> them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple
> iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format,
> while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

> Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute
> power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each
> cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies
> at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better
> training and model accuracy than the IEEE half-precision representation.

Single-precision floating-point format

From Wikipedia, the free encyclopedia

> **Single-precision floating-point format** is a computer number format, usually occupying 32 bits in computer
> memory; it represents a wide dynamic range of numeric values by using a floating radix point.
> A floating-point variable can represent a wider range of numbers than a fixed-point variable of the same bit width at

f.   The "*first input signal*" and the "*second input* signal" for each individual MXU

Reduced Precision Multiply Cell are respectively the signals representing the two

float32 input values before they are converted to bfloat16 format and multiplied

by the MXU Multiplier Circuit. The "*first output signal*" produced by an MXU

Reduced Precision Multiply Cell is the result of the multiplication operation

performed on these float32 values after being reduced to bfloat16 precision. That

result of the multiplication operation, i.e., "*a third numerical value*," is

represented by the "*first output signal*." As published by Google:

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

On a GPU, one would program this dot product into a GPU "core" and then execute it on as many "cores" as are available in parallel to try and compute every value of the resulting matrix at once. If the resulting matrix is 128x128 large, that would require 128x128=16K "cores" to be available which is typically not possible. The largest GPUs have around 4000 cores. A TPU on the other hand uses the bare minimum of hardware for the compute units in the MXU: just `bfloat16 x bfloat16 => float32` multiply-accumulators, nothing else. These are so small that a TPU can implement 16K of them in a 128x128 MXU and process this matrix multiplication in one go.

Illustration: the MXU systolic array. The compute elements are multiply-accumulators. The values of one matrix are loaded into the array (red dots). Values of the other matrix flow through the array (grey dots). Vertical lines propagate the values up. Horizontal lines propagate partial sums. It is left as an exercise to the user to verify that as the data flows through the array, you get the result of the matrix multiplication coming out of the right side.

Let's see how a systolic array executes the neural network calculations. At first, the TPU loads the parameters from memory into the matrix of multipliers and adders.
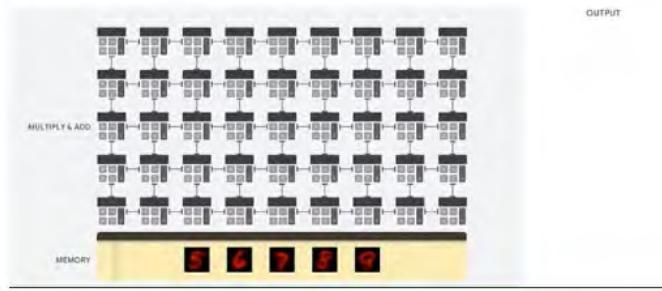
Then, the TPU loads data from memory. As each multiplication is executed, the result will be passed to the next multipliers while taking the summation at the same time. So the output will be the summation of all multiplication results between data and parameters. During the whole process of massive calculations and data passing, no memory access is required at all.
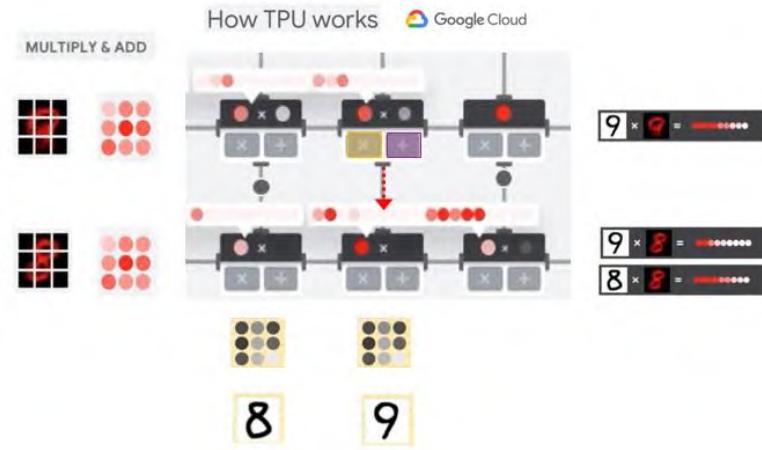
53.   In the Accused TPU device, each one of first execution units "*has access to memory local to that execution unit.*" Each of the aforementioned first execution units (an MXU Reduced Precision Multiply Cell) has an associated memory unit, as shown in Figure 3 of the Google patent application 2018/0336165 ("the Google '165 patent application"), whose specification has been represented by Google as being reflective of the architecture of the TPUv2 chip and/or the TPUv3 chip, which as noted before, has features common with at least one of the Accused TPU Devices. Each such memory unit is local to its associated processing element. *See also*, *e.g.*, https://cloud.google.com/tpu/docs/beginners-guide ("the TPU loads the parameters from memory into the matrix of multipliers and adders").

35

FIG. 3
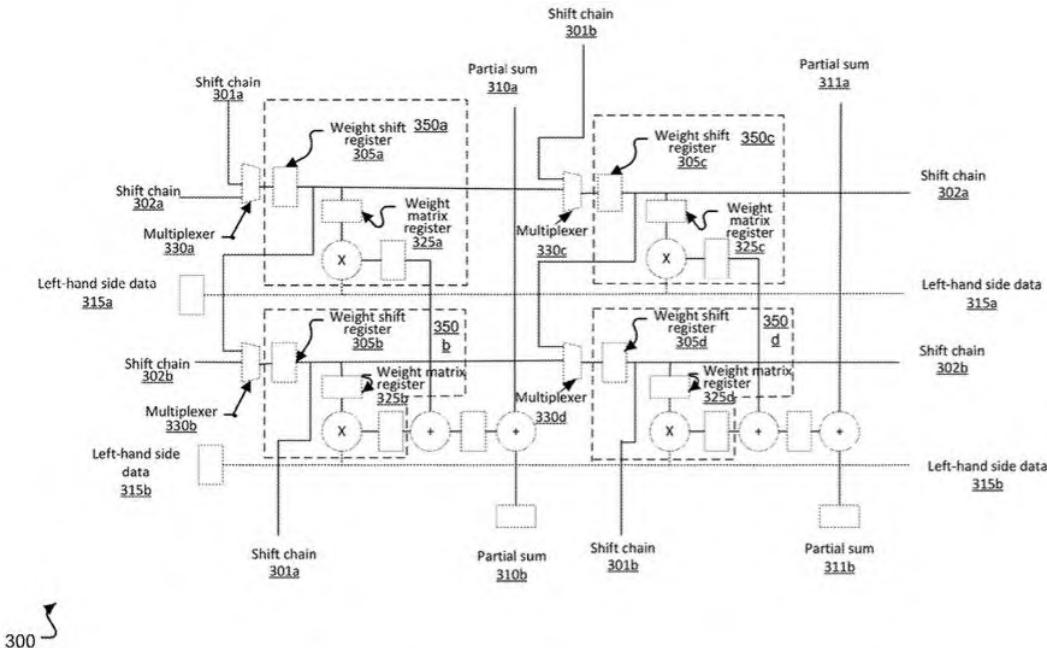
This memory is used, for example, to store "weights" or "parameters" as part of algorithms that relate to neural networks. *See*, *e.g.*, https://www.programmersought.com/article/66614714332/ and previously https://cloud.google.com/tpu/docs/beginners-guide ("the TPU loads the parameters from memory into the matrix of multipliers and adders").

54.   In the Accused TPU device, each execution unit "*produces one or more first output signals that represent a third numerical value, wherein the dynamic range of possible valid inputs to the first operation is at least as wide as from 1/1,000,000,000 through 1,000,000,000 and for each of at least X=10% of the possible valid inputs to the first operation the numerical value represented by the one or more first output signals differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that input.*" Specifically:

   a.   For each MXU Reduced Precision Multiply Cell (the "*execution unit*"), "*the dynamic range of the possible valid inputs to the first operation is at least as wide*

36

as from 1/1,000,000 through 1,000,000." As shown above, each MXU Reduced

Precision Multiply Cell performs a float32 multiplication operation at "reduced

bfloat16 precision" on valid input signals representing numerical values having a

float32 format. A float32 numerical value, whose format is shown below, has the

following dynamic range:

Minimum: $2^{-126} \approx 1.175494351 \times 10^{-38}$

Maximum: $(2 - 2^{-23}) \times 2^{127} \approx 3.402823466 \times 10^{38}$

Even after the convertion of each float32 input into a respective bfloat15 input, the dynamic

range of each such input is still approximately the same, from about 1×10^{-38} to about 3×10^{-38}.

As published by Google:

(a) fp32: Single-precision IEEE Floating Point Format — Range: ~1e⁻³⁸ to ~3e³⁸

Exponent: 8 bits   Mantissa (Significand): 23 bits

(b) fp16: Half-precision IEEE Floating Point Format — Range: ~5.96e⁻⁸ to 65504

Exponent: 5 bits   Mantissa (Significand): 10 bits

(c) bfloat16: Brain Floating Point Format — Range: ~1e⁻³⁸ to ~3e³⁸

Exponent: 8 bits   Mantissa (Significand): 7 bits
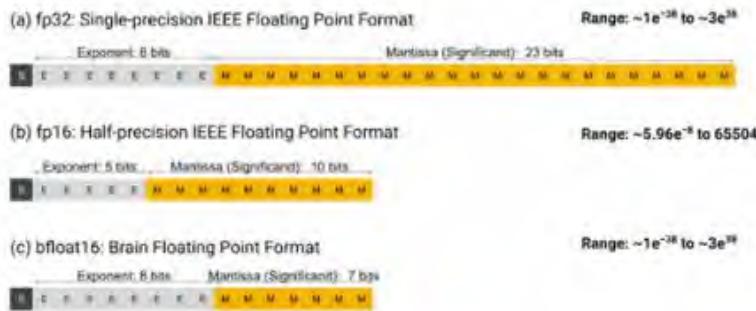
As Figure 1 shows, bfloat16 has a greater dynamic range—i.e., number of exponent
bits—than FP16. In fact, the dynamic range of bfloat16 is identical to that of FP32.
We've trained a wide range of deep learning models, and in our experience, the
bfloat16 format works as well as the FP32 format while delivering increased
performance and reducing memory usage.

b.   For each MXU Reduced Precision Multiply Cell, "*for at least X=10% of the*

possible valid inputs to the first operation the numerical values represented by the

one or more first output signal of the LPHDR unit executing the first operation on

that input differs by at least Y=0.05% from the result of an exact mathematical

calculation of the first operation on the numerical values of that input."

Specifically, each MXU Reduced Precision Multiply Cell performs a float32

multiplication operation but does so in Google's own words at "reduced bfloat16

precision." As published by Google:

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication
unit (MXU),   a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and
multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the
majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then
automatically converts them to bfloat16 before calculation. (A TPU can perform
FP32 multiplications via multiple iterations of the MXU.) Inside the MXU,
multiplications are performed in bfloat16 format, while accumulations are
performed in full FP32 precision.

Cloud TPU
System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute
power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each
cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies
at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better
training and model accuracy than the IEEE half-precision representation.

c.   Each MXU Reduced Precision Multiply Cell takes the following steps: (i)

receives as input two signals that each represent a float32 numerical value, (ii)

converts each of the received float32 numerical values to a bfloat16 numerical

value, (iii) multiplies the resulting pair of bfloat16 numerical values with each

other, and (iv) adjusts the format of the result of the bfloat16 multiplication

generated in step (iii), if needed, to produce an output signal that represents a

float32 numerical value to be accumulated. When the float32 numerical values

that are output by the Accused TPU Device's low precision multiplication

operations, (i.e., the multiplication operation performed on float32 inputs at

"reduced bfloat16 precision"), are compared to the numerical values that would

have been produced by the exact full precision multiplication operations on those

same respective valid input float32 numerical values (the ones mentioned in (i)

just above), the Accused TPU Device's output float32 numerical values differ, for

at least 10% of those input float32 numerical values, from the respective values

that would have been produced by the exact full precision multiplication

operations, by at least 0.05%. This is illustrated by the Singular test results shown

below.

		bf16
% of valid > 1.00%		4.65%
% of valid > 0.50%		55.39%
% of valid > 0.20%		92.69%
% of valid > 0.10%		98.15%
% of valid > 0.05%		99.52%

55.   In each Accused TPU Device, there is at least one "*second execution unit adapted*

to execute a second operation of traditional high-precision multiplication on floating point

numbers that are at least 32 bits wide." Each TensorCore (each TPUv4 has two TensorCores and

TPUv5e has a TensorCore, as explained above) has a Vector Processing Unit (VPU). Each VPU

has 2,048 (16 x 128) ALUs, half of which are execution units adapted to execute multiplication

on floating point numbers that are at least 32 bits wide. *See*, *e.g.*,

https://codelabs.developers.google.com/codelabs/keras-flowers-data/#2 ("The VPU handles

float32 and int32 computations." As published by Google:

> Figure 2 below shows a TPU v4 package and four of them
> mounted on the printed circuit board. Like TPU v3, each TPU v4
> contains two *TensorCores* (*TC*). Each TC contains four 128x128
> *Matrix Multiply Units* (*MXUs*) and a Vector *Processing Unit* (*VPU*)
> with 128 lanes (16 ALUs per lane) and a 16 MiB *Vector Memory*
> (*VMEM*). The two TCs share a 128 MiB Common Memory

[0046] The computational unit includes vector registers, i.e., **32** vector registers, in a vector processing unit (**106**) that can be used for both floating point operations and integer operations. The computational unit includes two arithmetic logic units (ALUs) (**126***c-d*) to perform computations. One ALU (**126***c*) performs floating point addition and the other ALU (**126***d*) performs floating point multiplication. Both

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

56.   In the Accused TPU *Devices, "a total number of first execution units in the silicon chip exceeds, by at least 100 more than five times, a total number of execution units in the silicon chip adapted to execute the operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide;"* Each MXU Reduced Precision Multiply Cell is one of the first execution units, as described above. A TPUv4 chip has eight MXUs (two TensorCores per TPUv4, and four MXUs per TensorCore), while a TPUv5 chip has four MXUs (one TensorCores per TPUv5e, and four MXUs per TensorCore). Each MXU has 16,384 MXU Reduced Precision Multiply Cells as shown above. Therefore, a TPUv4 chip has 131,072 MXU Reduced Precision Multiply Cells and a TPUv5 chip has 65,536 MXU Reduced Precision Multiply Cells, which each are the "*first execution units.*" By contrast, as shown above, each TensorCore has a VPU, each VPU has 2,048 ALUs, and each VPU has 1,024 ALUs that perform traditional high precision multiplication on float32 numbers, which are the "*second execution units*" in the claim. Therefore, each TPUv4 chip has 131,072 of the first execution units and 2,048 of the second execution units (from two TensorCores). Each TPUv5 chip has 65,536 of the first execution units and 1,024 of the second execution units (from one TensorCore). 131,072 exceeds by at least 100 more than five times 2,048, and 65,536 exceeds by at least 100 more than

five times 1,024. Therefore, for all the Accused TPU Devices, there are "*a total number of first*

execution units in the silicon chip exceeds, by at least 100 more than five times, a total number of

execution units in the silicon chip adapted to execute the operation of traditional high-precision

multiplication on floating point numbers that are at least 32 bits wide." As published by Google:

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

57.   In the Accused TPU Devices, "*each of the plurality of first execution units is*

smaller than the second execution unit." MXU Reduced Precision Multiply Cells as defined

above are smaller than the VPU ALUs that are multipliers. Google engineer Jeffrey Dean, the

head of Google Brain, expressly admitted this:

> Furthermore, one major area & power cost of multiplier circuits for a
> floating point format with M mantissa bits is the $(M+1) \times (M+1)$ array of
> full adders (that are needed for multiplying together the mantissa portions
> of the two input numbers. The IEEE fp32, IEEE fp16 and bfloat16
> formats need **_576 full adders_**, 121 full adders, and **_64 full adders_**,
> respectively. **_Because multipliers for the bfloat16 format require so_**
> **_much less circuitry_**, it is possible to put more multipliers in the same chip
> area and power budget, thereby meaning that ML accelerators employing
> this format can have higher flops/sec and flops/Watt, all other things being
> equal.

Dean, Jeffrey. (2020). 1.1 *The Deep Learning Revolution and Its Implications for Computer*

Architecture and Chip Design. 8-14. 10.1109/ISSCC19947.2020.9063049 (emphasis added).

This fact was further confirmed in a paper published by the team of Google engineers

responsible for designing and building the accused TPUs (including, *inter alia*, Norman Jouppi

and David Patterson):

Operation		Picojoules per Operation		
		45 nm	7 nm	45 / 7
+	Int 8	0.03	0.007	4.3
	Int 32	0.1	0.03	3.3
	BFloat 16	--	0.11	--
	IEEE FP 16	0.4	0.16	2.5
	IEEE FP 32	0.9	0.38	2.4
×	Int 8	0.2	0.07	2.9
	Int 32	3.1	1.48	2.1
	BFloat 16	--	0.21	--
	IEEE FP 16	1.1	0.34	3.2
	IEEE FP 32	3.7	1.31	2.8
SRAM	8 KB SRAM	10	7.5	1.3
	32 KB SRAM	20	8.5	2.4
	1 MB SRAM[1]	100	14	7.1
GeoMean[1]		--	--	2.6
		Circa 45 nm	Circa 7 nm	
DRAM	DDR3/4	1300[2]	1300[2]	1.0
	HBM2	--	250-450[2]	--
	GDDR6	--	350-480[2]	--

Table 2. Energy per Operation: 45 nm [16] vs 7 nm. Memory is pJ per 64-bit access.

Jouppi, Norman, *et al.*. "Ten Lessons From Three Generations Shaped Google's TPUv4i: Industrial Product," in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, Valencia, Spain, 2021 pp. 1-14 at 3 (emphasis added). According to the above table, a BFloat 16 multiplier requires less than 20% as much energy per operation as an IEEE FP32 multiplier (both made using the same 7 nm semiconductor fabrication process). The lower power requirements of BFloat16 multipliers is a result of the fact that they include fewer transistors than full-precision IEEE FP32 multipliers.

58.   In the Accused TPU Devices, there is "*decoding circuitry adapted to decode the at least one instruction received from the at least one instruction memory and to send at least one control signal to at least one of the plurality of first execution units to cause the at least one of the plurality of first execution units to operate according to the at least one instruction.*" As shown above, the Accused TPU Devices comprise instruction memory and also comprise circuitry for executing the instructions (for example the TPU arithmetic operations described above). Therefore, the Accused TPU Devices have such decoding circuitry.
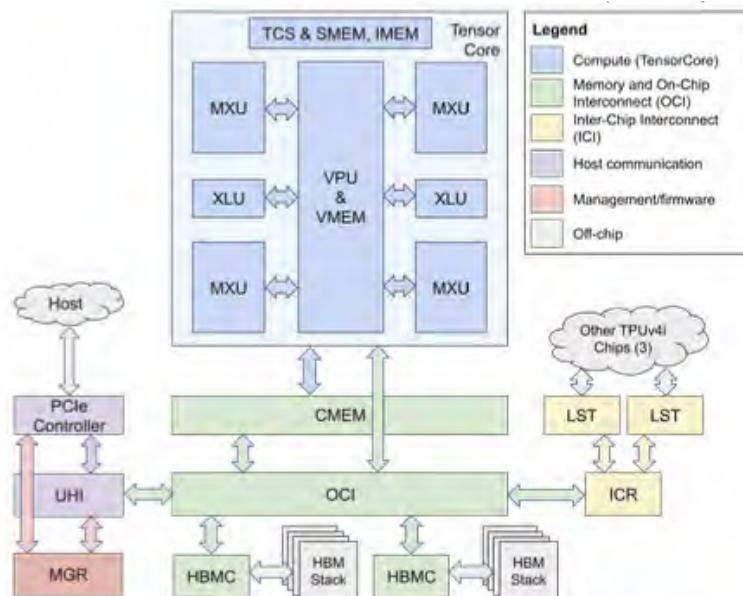
42

FIG. 1C

Figure 5. TPUv4i chip block diagram. Architectural memories are HBM, Common Memory (CMEM), Vector Memory (VMEM), Scalar Memory (SMEM), and Instruction Memory (IMEM). The data path is the Matrix Multiply Unit (MXU), Vector Processing Unit (VPU), Cross-Lane Unit (XLU), and TensorCore Sequencer (TCS). The uncore (everything not in blue) includes the On-Chip Interconnect (OCI), ICI Router (ICR), ICI Link Stack (LST), HBM Controller (HBMC), Unified Host Interface (UHI), and Chip Manager (MGR).

43

> The VLIW instruction needed extra fields to handle the four MXUs and the CMEM scratchpad memory, which were easy to add given no need for binary compatibility. The TPUv4i instruction is 25% wider than TPUv3.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

59.   In the Accused TPU Devices, "*the plurality of first execution units are adapted to collectively perform, per cycle, at least tens of thousands of the first operation*." The first execution units are the MXU Reduced Precision Multiply Cells as described above. The first operation is the multiplication operation being performed at "reduced bfloat16 precision" by each MXU Reduced Precision Multiply Cell. Collectively, the MXU Reduced Precision Multiply Cells in TPUv4 chips perform at least tens of thousands of first bfloat16 multiplication operations per clock cycle (138 bfloat16 TFLOPS with a clock rate of 1050MHz, which means the Accused TPU Device is performing $\approx$ 131,000 reduced precision bfloat16 multiplication operations per clock cycle, and since TPUv5 chips have half the MXUs as TPUv4 chips, the MXU Reduced Precision Multiply Cells in TPUv5 chips perform $\approx$ 65,000 bfloat16 reduced precision multiplication operations per clock cycle).

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

Feature	TPUv1	TPUv2	TPUv3	TPUv4i	NVIDIA T4
Peak TFLOPS / Chip	92 (8b int)	46 (bf16)	123 (bf16)	138 (bf16/8b int)	65 (ieee fp16)/130 (8b int)
First deployed (GA date)	Q2 2015	Q3 2017	Q4 2018	Q1 2020	Q4 2018
DNN Target	Inference only	Training & Inf.	Training & Inf.	Inference only	Inference only
Network links x Gbits/s / Chip	--	4 x 496	4 x 656	2 x 400	--
Max chips / supercomputer	--	256	1024	--	--
Chip Clock Rate (MHz)	700	700	940	1050	585 / (Turbo 1590)
Idle Power (Watts) Chip	28	53	84	55	36
TDP (Watts) Chip / System	75 / 220	280 / 460	450 / 660	175 / 275	70 / 175
Die Size (mm^2)	< 330	< 625	< 700	< 400	545
Transistors (B)	3	9	10	16	14
Chip Technology	28 nm	16 nm	16 nm	7 nm	12 nm
Memory size (on-/off-chip)	28MB / 8GB	32MB / 16GB	32MB / 32GB	144MB / 8GB	18MB / 16GB
Memory GB/s / Chip	34	700	900	614	320 (if ECC is disabled)
MXU Size / Core	1 256x256	1 128x128	2 128x128	4 128x128	8 8x8
Cores / Chip	1	2	2	1	40
Chips / CPUHost	4	4	4	8	8

Table 1. Key characteristics of DSAs. The underlines show changes over the prior TPU generation, from left to right. System TDP includes power for the DSA memory system plus its share of the server host power, e.g., add host TDP/8 for 8 DSAs per host.

60.   In knowingly adopting Dr. Bates' patented computer architectures, Google reaps the very same benefits that were predicted by Dr. Bates in his patent application more than 10 years ago. As published by Google and predicted by Dr. Bates in his patent application:

Choosing bfloat16

Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency while maintaining the ability to train accurate deep learning models, all with minimal switching costs from FP32. The physical size of a hardware multiplier scales with the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16 multipliers are about half the size in silicon of a typical FP16 multiplier, and they are *eight times* smaller than an FP32 multiplier!

PEs implemented according to certain embodiments of the present invention may be relatively small for PEs that can do arithmetic. This means that there are many PEs per unit of resource (e.g., transistor, area, volume), which in turn means that there is a large amount of arithmetic computational power per unit of resource. This enables larger problems to be solved with a given amount of resource than does traditional computer designs. For instance, a digital embodiment of the present invention built as a large silicon chip fabricated with current state of the art technology might perform tens of thousand of arithmetic operations per cycle, as opposed to hundreds in a conventional GPU or a handful in a conventional multicore CPU. These ratios reflect an architectural advantage of embodiments of the present invention that should persist as fabrication technology continues to improve, even as we reach nanotechnology or other implementations for digital and analog computing.

61.   Due to its monitoring of Singular's patents and applications, Google knew of the application for the '714 patent prior to the issuance of the patent on May 10, 2022. For example,

Google's attorneys prepared and filed two petitions for *Inter Partes* Review ("IPR") of the '616

patent. In each of those petitions, Google identified numerous patents and applications related to

the '616 patent, including application serial number US17/367,051, which led to the '714 patent.

Thus, at least since 12/22/2022, when Google identified the '051 application in, *inter alia*, its

Petition for *Inter Partes* Review in IPR2023-00395, Google has knowledge of the '051

application. Before making such identification, counsel for Google reviewed application serial

number US17/367,051.

62.   As a result of Google's infringement of the '714 patent, Singular has suffered

damages in an amount to be determined at trial.

COUNT II
(Google's Infringement of United States Patent No. 10,754,616)

63.   Paragraphs [1-62] are reincorporated by reference as if fully set forth herein.

64.   Google has directly infringed, and continues to directly infringe, literally and/or

by the doctrine of equivalents, at least claim 7 of the '616 patent by making, using, testing,

selling, offering for sale and/or importing into the United States the Accused TPU Computing

Systems alone or in combination with its existing data servers. An Accused TPU Computing

System include a plurality of Accused TPU Devices. Cloud TPU (an Accused TPU Computing

System), in Google's own words, "powers" at least Google Translate, Photos, Search, Assistant,

and Gmail. As published by Google:

Empowering businesses with Google Cloud AI

Machine learning has produced business and research breakthroughs ranging from network security to medical diagnoses. We built the Tensor Processing Unit (TPU) in order to make it possible for anyone to achieve similar breakthroughs. Cloud TPU is the custom-designed machine learning ASIC that powers Google products like Translate, Photos, Search, Assistant, and Gmail. Here's how you can put the TPU and machine learning to work accelerating your company's success, especially at scale.

65.   The Accused TPU Computing Systems are each examples of a "*computing system,*" as claimed by the '616 patent. As published by Google:

Cloud TPU > Documentation > Guides   Was this helpful? 👍 👎

System Architecture 🔖 ▾   Send feedback

Tensor Processing Units (TPUs) are application specific integrated circuits (ASICs) designed by Google to accelerate machine learning workloads. Cloud TPU is a Google Cloud service that makes TPUs available as a scalable resource.

TPUs are designed to perform matrix operations quickly making them ideal for machine learning workloads. You can run machine learning workloads on TPUs using frameworks such as TensorFlow, Pytorch, and JAX.

47

TPU VM Architecture

The TPU VM architecture lets you directly connect to the VM physically connected to the TPU device using SSH. You have root access to the VM, so you can run arbitrary code. You can access compiler and runtime debug logs and error messages.

66.   The Accused TPU Computing Systems each comprise a "*host computer*" as claimed by the '616 patent. Each TPU board, which each have multiple Accused TPU Devices, is connected to a TPU Host—a physical computer connected to one or more TPU boards on which there are Accused TPU Devices, on which a virtual machine (VM) runs—for loading and preprocessing data to be fed to the Accused TPU Devices. The TPU Host is a "*host computer.*"

48

Single host and multi host 🔗

A TPU host is a VM that runs on a physical computer connected to TPU hardware. TPU workloads can use one or more host.

A single-host workload is limited to one TPU VM and can access 1, 4, or 8 TPU chips. A multi-host TPU v5e workload can access 8, 12, 16, 32, 64, 128, or 256 TPU chips with one TPU VM for every four TPU chips. Multi-host workloads distribute training across multiple TPU VMs.

TPU v5e supports single and multi-host training and single host inference. Multi-host inference is supported using Sax. For more information, see Large Language Model Serving.

Cloud TPU VM Architectures

How you interact with the TPU host (and the TPU board) depends upon the TPU VM architecture you're using: TPU Nodes or TPU VMs.

TPU Node Architecture

★ **Important:** TPU v4 is not supported with the TPU Node architecture.

The TPU Node architecture consists of a user VM that communicates with the TPU host over gRPC. When using this architecture, you cannot directly access the TPU Host, making it difficult to debug training and TPU errors.

TPU VM Architecture

The TPU VM architecture lets you directly connect to the VM physically connected to the TPU device using SSH. You have root access to the VM, so you can run arbitrary code. You can access compiler and runtime debug logs and error messages.

67.   The Accused TPU Computing System comprises the Accused TPU Devices, each of which is a "*computing chip*."

Cloud TPU > Documentation > Guides   Was this helpful?

System Architecture   Send feedback

Tensor Processing Units (TPUs) are application specific integrated circuits (ASICs) designed by Google to accelerate machine learning workloads. Cloud TPU is a Google Cloud service that makes TPUs available as a scalable resource.

TPUs are designed to perform matrix operations quickly making them ideal for machine learning workloads. You can run machine learning workloads on TPUs using frameworks such as TensorFlow, Pytorch, and JAX.

50

TPU chip

A TPU chip contains one or more TensorCores. The number of TensorCores depend on the version of the TPU chip. Each TensorCore consists of one or more matrix-multiply units (MXUs), a vector unit, and a scalar unit.

68.   In the Accused TPU Computing System, the Accused TPU Devices comprise "*a processing element array comprising a plurality of first processing elements, wherein the plurality of first processing elements is no less than 5000 in number, wherein each of a first subset of the plurality of first processing elements is positioned at a first edge of the processing element array, and wherein each of a second subset of the plurality of first processing elements is positioned in the interior of the processing element array.*" As published by Google:

a. The Accused TPU Devices each contain TensorCores which each have one of more MXUs. Specifically, each TPUv4 chip has 8 MXUs (four MXUs per TensorCore, 2 TensorCores per chip), and each TPUv5 chip has 4 MXUs (four MXUs per TensorCore, 1 TensorCore per chip). As published by Google:

TPU chip

A TPU chip contains one or more TensorCores. The number of TensorCores depend on the version of the TPU chip. Each TensorCore consists of one or more matrix-multiply units (MXUs), a vector unit, and a scalar unit.

An MXU is composed of 128 x 128 multiply-accumulators in a systolic array. MXUs provide the bulk of the compute power in a TensorCore. Each MXU is capable of performing 16K multiply-accumulate operations per cycle. All multiplies take bfloat16 inputs, but all accumulations are performed in FP32 number format.

The vector unit is used for general computation such as activations and softmax. The scalar unit is used for control flow, calculating memory addresses, and other maintenance operations.

TensorCores

TPU chips have one or two TensorCores to run matrix multiplication. Similar to v2 and v3 Pods, v5e has one TensorCore per chip. By contrast, v4 Pods have 2 TensorCores per chip. For more information about TensorCores, see ACM article.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The following table shows the key specifications for a v4 TPU Pod.

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

b. Each MXU contains a systolic array having 128 x 128 "multiply-accumulators," each of which includes a multiplier circuit, an adder circuit, and various data movement circuitry (MXU Multiply Add Cell). The array of 128 x 128 MXU Multiply Add Cells is a "*a processing element array comprising a plurality of first processing elements, wherein the plurality of first processing elements is no less than 5000 in number.*" Each MXU Multiply Add Cell is a "*first processing element.*" Figure 1C, Figure 2 and Figure 3 below are taken from the Google '165 patent application, whose specification has been represented by Google as being reflective of the architecture of the TPUv2 chip and/or the TPUv3 chip, which as noted before, has features common with at least one of the Accused TPU Devices. As represented in the Google '165 patent application, Figure 3 illustrates a "multi-cell inside a matrix multiply unit" of a TensorCore that includes an array of "multiply-add sub-units that can be grouped into multi-cells." Each multiply-add sub-unit, i.e., a MXU Multiply Add Cell, is a "*first processing element.*"

An MXU is composed of 128 x 128 multiply-accumulators in a systolic array. MXUs provide the bulk of the compute power in a TensorCore. Each MXU is capable of performing 16K multiply-accumulate operations per cycle. All multiplies take bfloat16 inputs, but all accumulations are performed in FP32 number format.

The primary task for TPUs is matrix processing, which is a combination of multiply and accumulate operations. TPUs contain thousands of multiply-accumulators that are directly connected to each other to form a large physical matrix. This is called a systolic array architecture. Cloud TPU v3, contain two systolic arrays of 128 x 128 ALUs, on a single processor.

52

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.


FIG. 1C

53

FIG. 2

FIG. 3

54

[0055] Each MXU may have 128 rows and 128 columns. An MXU can be divided into identical blocks, referred to as tiles. For example, an MXU can be divided into 32 tiles, each of which contain 32 rows by 16 columns. Each tile can further be divided into multiply-add sub unit cells. Each cell takes a vector data input operand, multiplies the operand by stored weights to obtain a result, and adds the result to a partial sum to produce a new partial sum. In some implementations, the sub-unit cells can be grouped into larger multi-cells, i.e., 2×2 arrays of multiply-add sub-unit cells or 4×4 arrays of multiply-add sub-unit cells, referred to as sedecim cells. Instead of moving input data from one multiply-add sub-unit cell to the next at a rate of one per clock cycle, the data can move across the systolic array at one multi-cell per clock cycle.

c. As described above, a TPUv4 chip has eight MXUs (two TensorCores per TPUv4, and four MXUs per TensorCore), while a TPUv5 chip has four MXUs (one TensorCores per TPUv5e, and four MXUs per TensorCore). As also described above, each MXU has 16,384 MXU Multiply Add Cells as shown above. Therefore, a TPUv4 chip has 131,072 MXU Multiply Add Cells and a TPUv5 chip has 65,536 MXU Multiply Add Cells, which each are the "*first processing elements.*" A computing chip in the Accused TPU System therefore has a "*plurality of first processing elements is no less than 5000 in number.*"

d. In the MXU, "*each of a first subset of the plurality of first processing elements is positioned at a first edge of the processing element array, and wherein each of a second subset of the plurality of first processing elements is positioned in the interior of the processing element array.*" Each MXU includes horizontally and vertically interconnected processing elements arranged in systolic arrays, with a first subset of the MXU Multiply Add Cells ("*first processing elements*") for example positioned in a column at the left edge of the array starting at the second row and ending at the sixth row, and a second subset of the MXU Multiply Add Cells ("*first processing*

55

elements") for example positioned interior and to the right of the at least five left edge processing elements:
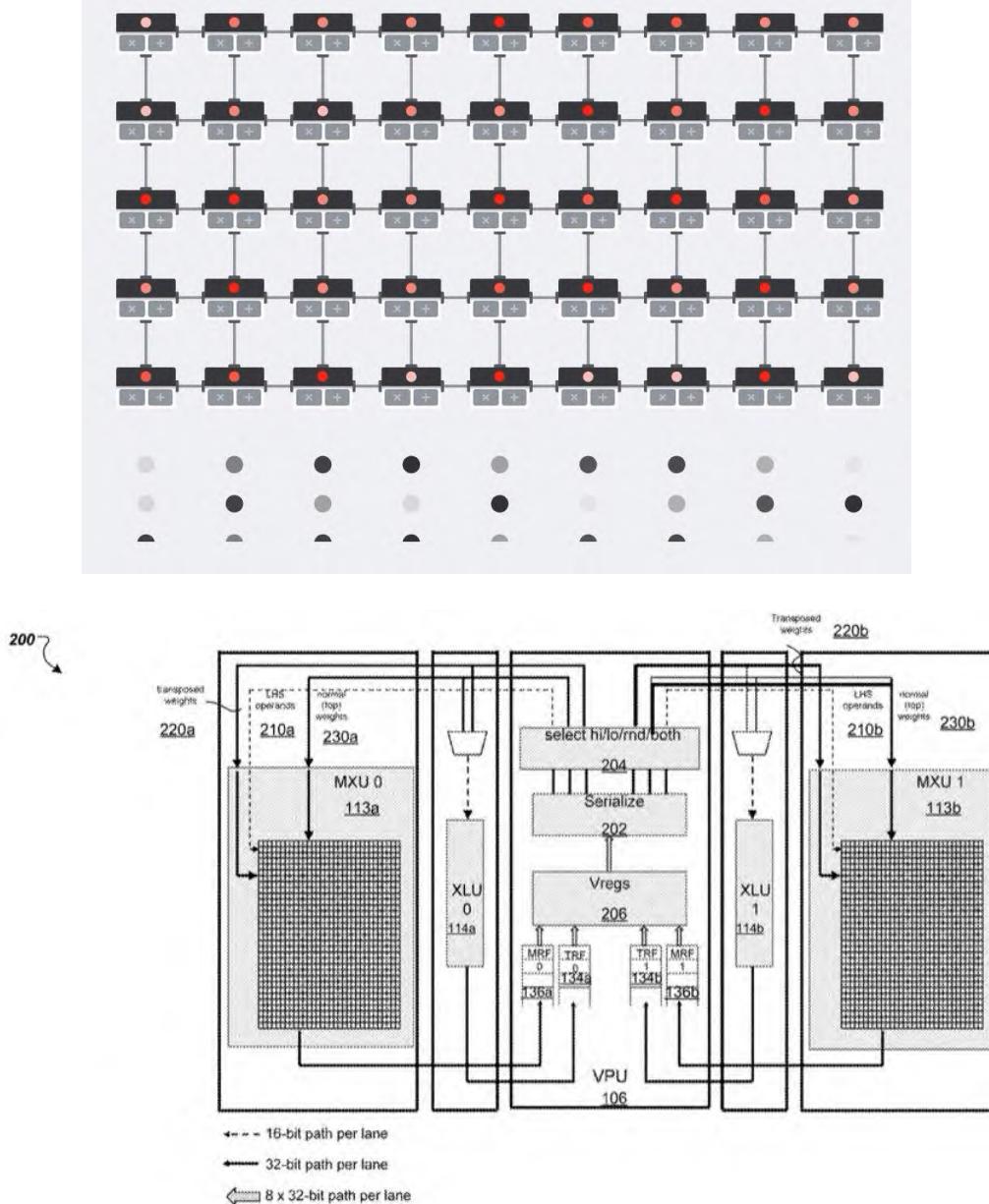
FIG. 2

69.   In the computing chip (an Accused TPU Device, which can be a TPUv4 chip or a TPUv5 chip) of the Accused TPU System, there is "*an input-output unit connected to each of the first subset of the plurality of first processing elements.*" As illustrated in Figure 2 and Figure 1C

56

of Google's 165 patent application as shown below, each TPU chip includes at least one input-output unit connected to the aforementioned first subset of MXU Multiply Add Cells ("*processing elements*") positioned at the edge of the array.
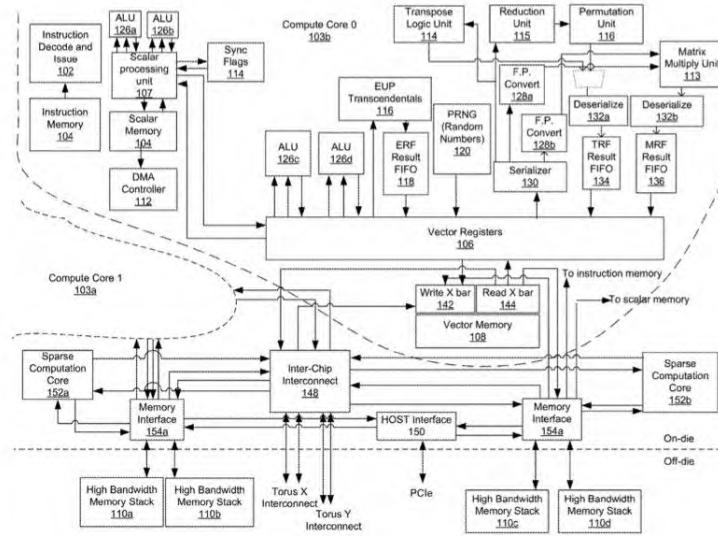
FIG. 1C
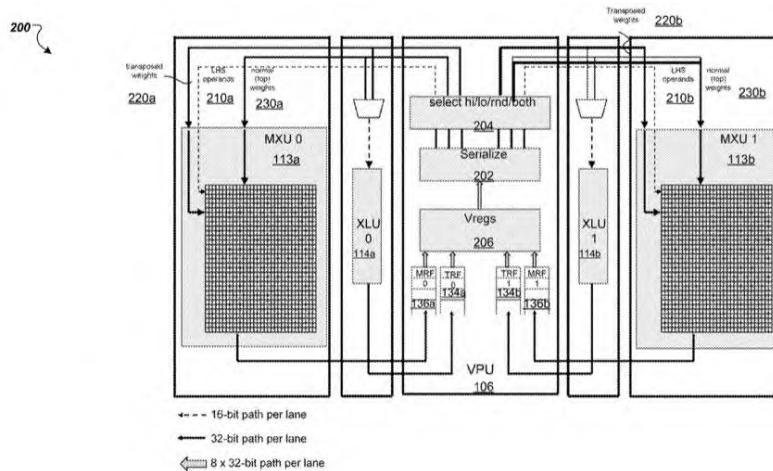
FIG. 2
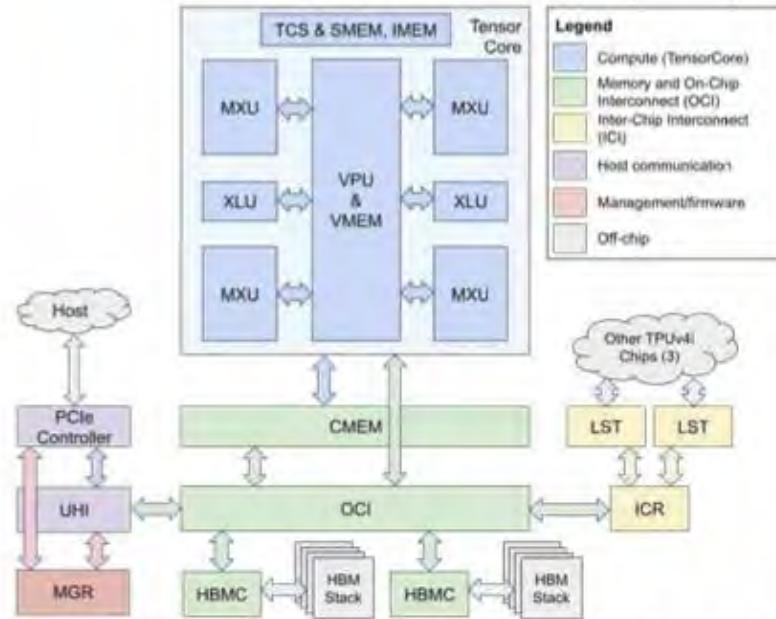
Figure 5. TPUv4i chip block diagram. Architectural memories are HBM, Common Memory (CMEM), Vector Memory (VMEM), Scalar Memory (SMEM), and Instruction Memory (IMEM). The data path is the Matrix Multiply Unit (MXU), Vector Processing Unit (VPU), Cross-Lane Unit (XLU), and TensorCore Sequencer (TCS). The uncore (everything not in blue) includes the On-Chip Interconnect (OCI), ICI Router (ICR), ICI Link Stack (LST), HBM Controller (HBMC), Unified Host Interface (UHI), and Chip Manager (MGR).

70.   In the computing chip (an Accused TPU Device, which can be a TPUv4 chip or a TPUv5 chip) of the Accused TPU System, there is "*a plurality of processing element connections, each processing element connection connecting one of the plurality of first processing elements with another of the plurality of first processing elements, wherein each of the plurality of first processing elements is connected to at least one other of the plurality of first processing elements by at least one of the plurality of processing element connections.*" The MXU in the Accused TPU Device each comprise a plurality of processing element connections. Each such processing element connection connects a first processing element (i.e., an MXU Multiply Add Cell) with at least one other first processing element (i.e., another MXU Multiply Add Cell). Each such first processing element is connected to at least another such first

58

processing element by a processing element connection. Figure 1C, Figure 2 and Figure 3 below

are taken from the Google '165 patent application. As represented in the Google '165 patent

application, Figure 3 illustrates a "multi-cell inside a matrix multiply unit" of a TensorCore, in

which the cells 350 together with the data movement circuitry, e.g., multiplexers, are each an

MXU Multiply Add Cell. Figure 3 also illustrates many instances of a "*processing element*

connection connecting one of the plurality of first processing elements with another of the

plurality of first processing elements, wherein each of the plurality of first processing elements is

connected to at least one other of the plurality of first processing elements by at least one of the

plurality of processing element connections," namely the connections that are between the MXU

Multiply Add Cells (in Figure 3) (*see* the Google '165 patent application, paragraph 0042,

paragraph 0055, paragraph 0063 and Figure 3).
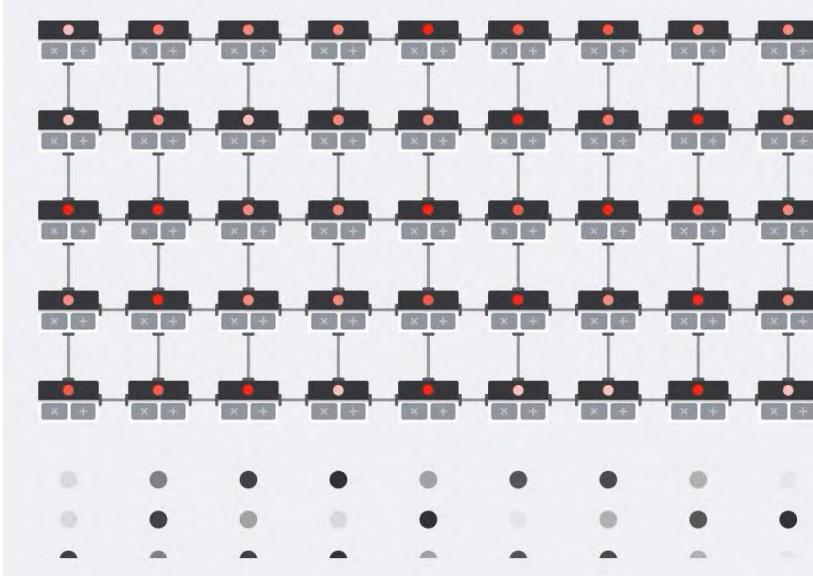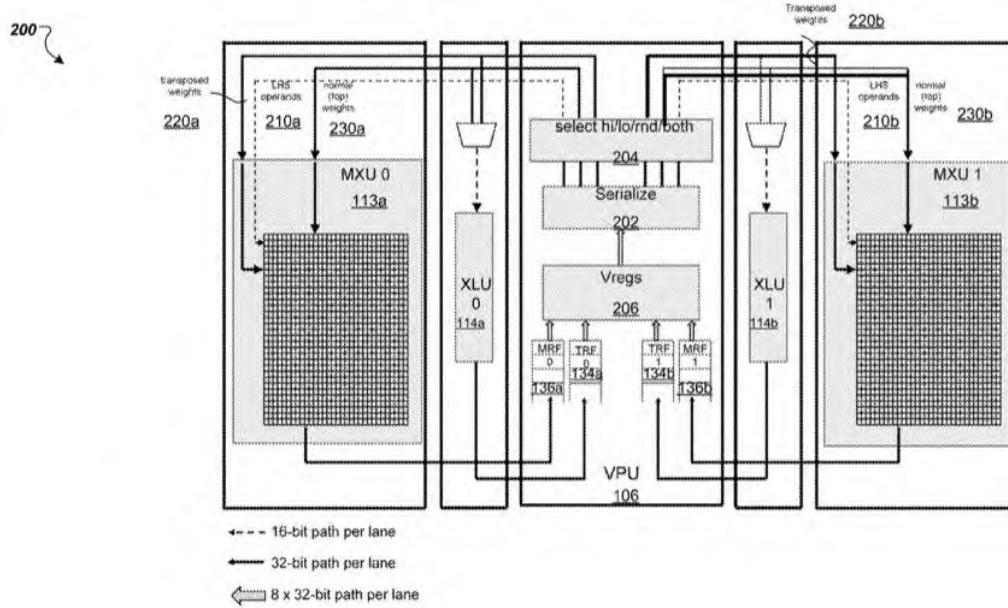
FIG. 2
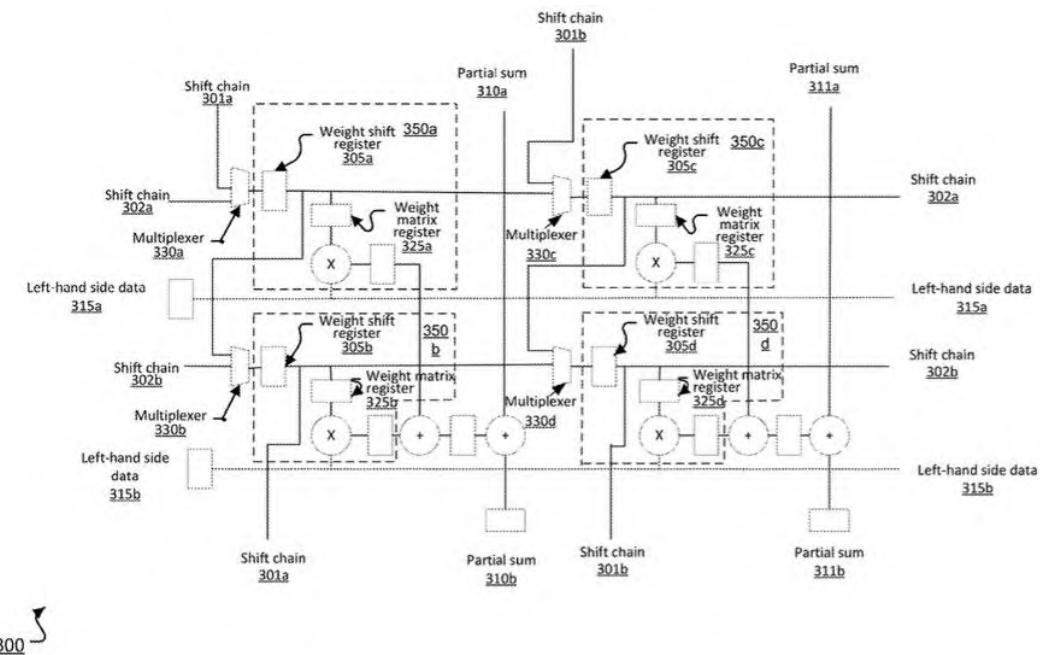
FIG. 3

71.　　In the computing chip (an Accused TPU Device, which can be a TPUv4 chip or a

TPUv5 chip) of the Accused TPU System, there is "*a plurality of memory units, wherein each of*

the plurality of first processing elements is associated with a corresponding one of the plurality

of memory units, and wherein each of the plurality of memory units is local to its associated one of the plurality of first processing elements." Each of the aforementioned processing elements (an MXU Multiply Add Cell) has an associated memory unit, as shown in Figure 3 of the Google '165 patent application. Each such memory unit is local to its associated processing element. *See also*, *e.g.*, https://cloud.google.com/tpu/docs/beginners-guide ("the TPU loads the parameters from memory into the matrix of multipliers and adders").
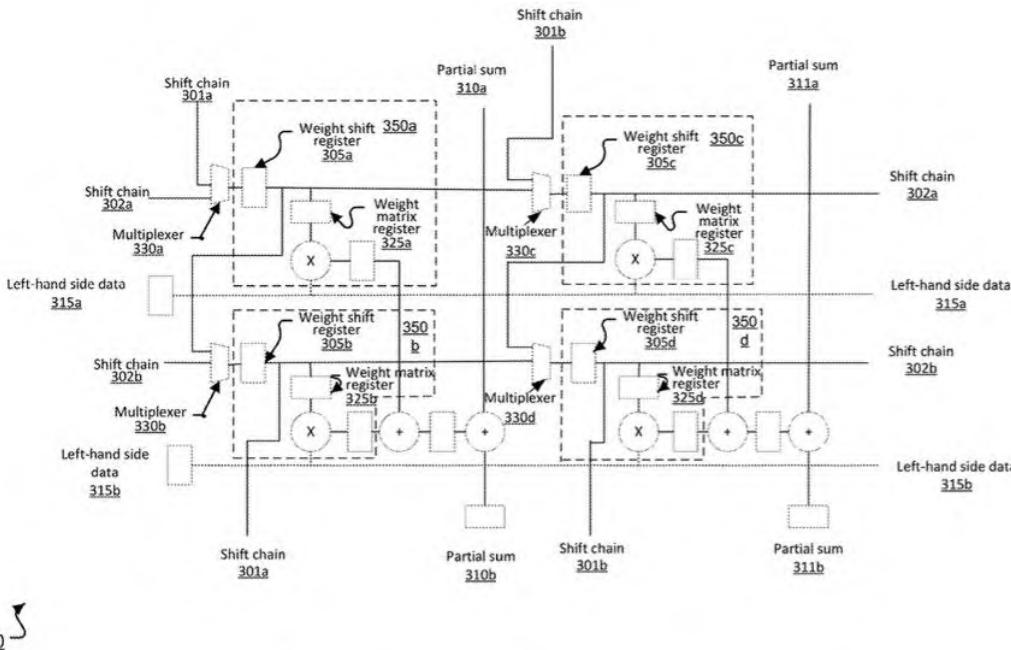
FIG. 3

72.   In the computing chip (an Accused TPU Device, which can be a TPUv4 chip or a TPUv5 chip) of the Accused TPU System, there is "*a plurality of arithmetic units, wherein each of the plurality of first processing elements has positioned therein at least one of the plurality of arithmetic units.*" As shown below in Figures 1c, 2 and 3 of the Google '165 patent application, each of the aforementioned first processing elements (each an MXU Multiply Add Cell) comprises a multiplier circuit and adder circuit (an "MXU AU"). An MXU AU is a one of "*a*

plurality of arithmetic units." An MXU AU in turn comprises a multiplier circuit (MXU Multiplier Circuit).
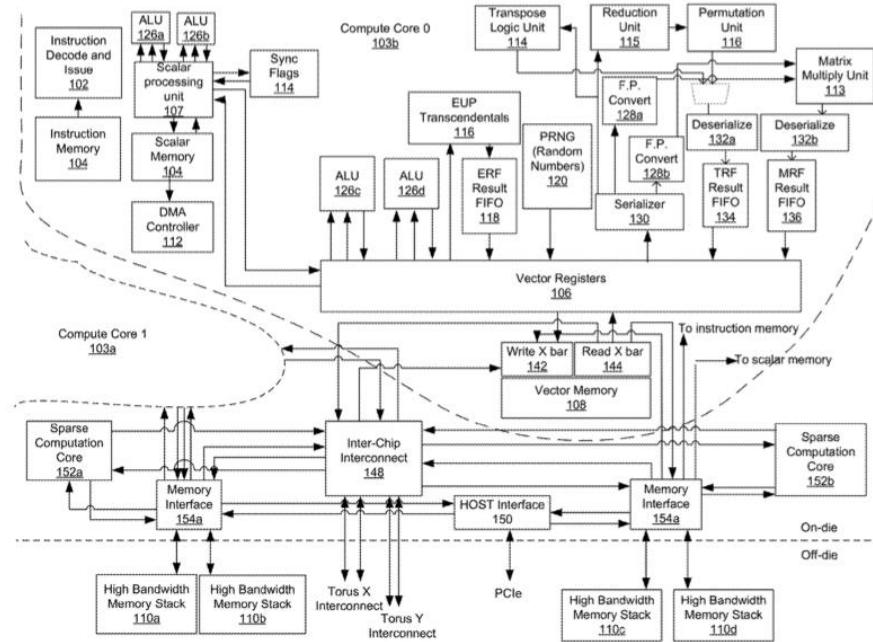
FIG. 1C
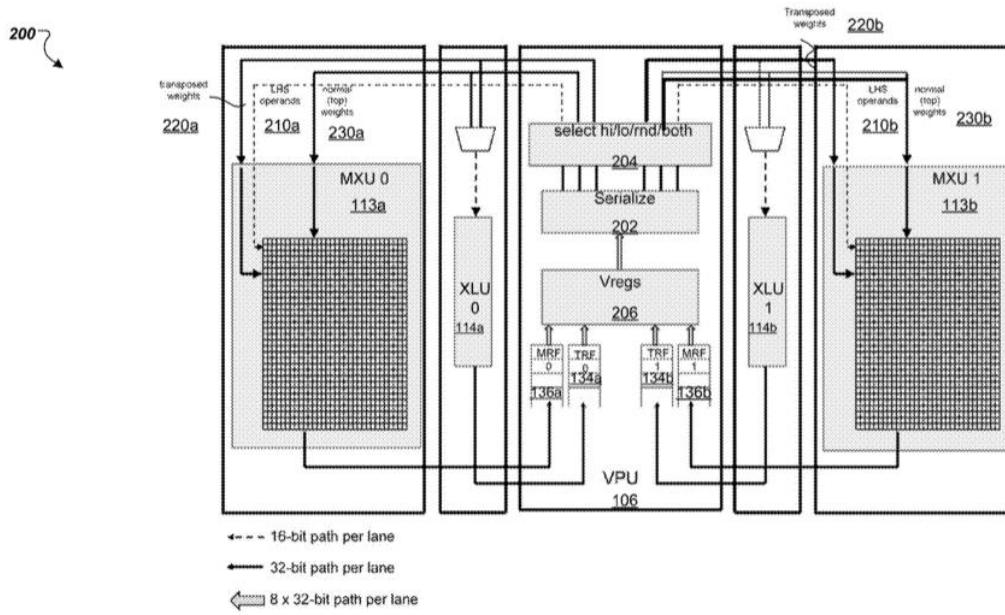
FIG. 2
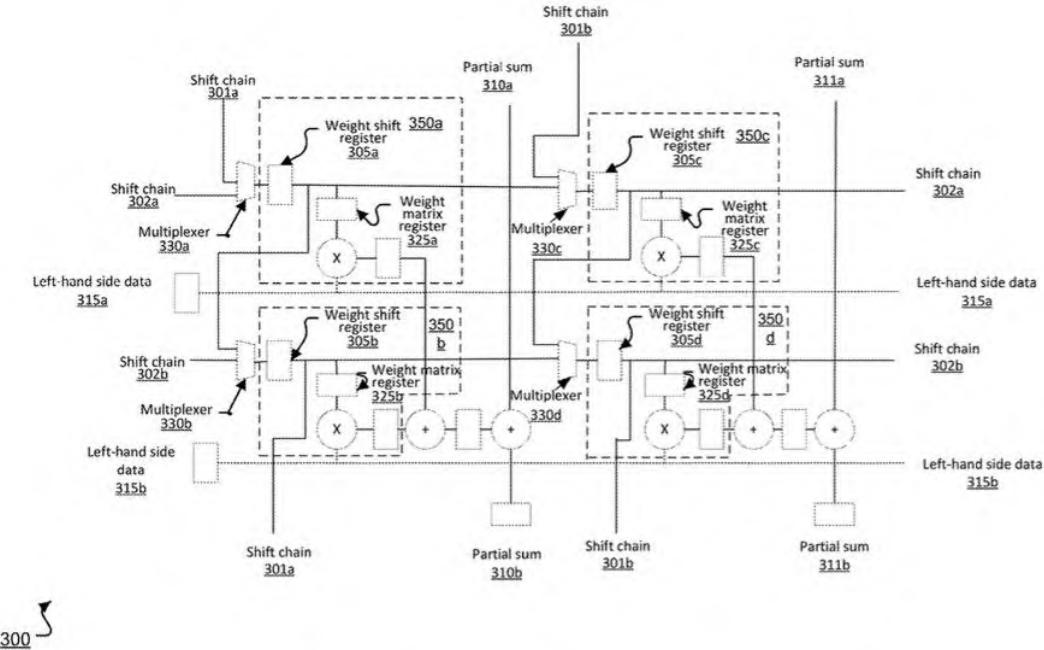
FIG. 3

73.   In the Accused TPU System, there is "*a host connection at least partially connecting the input-output unit with the host computer.*" The Accused TPU Devices—the TPUv4 chips and/or the TPUv5 chips—and the aforementioned input output units on those chips communicate with the TPU Host via host connections. *See* https://cloud.google.com/tpu/docs/. At least some of these host connections at least partially connect the aforementioned input-output units with the aforementioned TPU Host.

Single host and multi host 🔗

A TPU host is a VM that runs on a physical computer connected to TPU hardware. TPU workloads can use one or more host.

A single-host workload is limited to one TPU VM and can access 1, 4, or 8 TPU chips. A multi-host TPU v5e workload can access 8, 12, 16, 32, 64, 128, or 256 TPU chips with one TPU VM for every four TPU chips. Multi-host workloads distribute training across multiple TPU VMs.

TPU v5e supports single and multi-host training and single host inference. Multi-host inference is supported using Sax. For more information, see Large Language Model Serving.

Cloud TPU VM Architectures

How you interact with the TPU host (and the TPU board) depends upon the TPU VM architecture you're using: TPU Nodes or TPU VMs.

TPU Node Architecture

⭐ **Important:** TPU v4 is not supported with the TPU Node architecture.

The TPU Node architecture consists of a user VM that communicates with the TPU host over gRPC. When using this architecture, you cannot directly access the TPU Host, making it difficult to debug training and TPU errors.

TPU VM Architecture

The TPU VM architecture lets you directly connect to the VM physically connected to the TPU device using SSH. You have root access to the VM, so you can run arbitrary code. You can access compiler and runtime debug logs and error messages.

FIG. 1C

74.   In the Accused TPU System, and specifically in the Accused TPU Devices within

the Accused TPU System, the MXU AUs (the "*arithmetic units*") "*each comprises a first*

corresponding multiplier circuit adapted to receive as a first input to the first corresponding

multiplier circuit a first floating point value having a first binary mantissa of width no more than

11 bits and a first binary exponent of width at least 6 bits, and to receive as a second input to the

first corresponding multiplier circuit a second floating point value having a second binary

mantissa of width no more than 11 bits and a second binary exponent of width at least 6 bits."

The inputs received by the multiplier circuit (the MXU Multiplier Circuit) within each MXU

65

AU, comprise two floating point values having a bfloat16 format. Each MXU Multiplier Circuit performs the operation of multiplication on inputs each having a bfloat16 format. The bfloat16 format used in MXU Multiplier Circuits has 8 signed exponent bits and 8 signed mantissa bits, which means the "*multiplier circuits*" of the Accused TPU System are adapted to receive inputs having a binary mantissa of width that is no more than 11 bits and a binary exponent of width that is at least 6 bits.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Choosing bfloat16

Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency while maintaining the ability to train accurate deep learning models, all with minimal switching costs from FP32. The physical size of a hardware multiplier scales with the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16 multipliers are about half the size in silicon of a typical FP16 multiplier, and they are *eight times* smaller than an FP32 multiplier!

(c) bfloat16: Brain Floating Point Format   Range: ~$1e^{-38}$ to ~$3e^{38}$

Exponent: 8 bits   Mantissa (Significand): 7 bits

S E E E E E E E E M M M M M M M

75.   Google copied from Dr. Bates the idea of providing in a computing system thousands of processing elements in a processing element array that each perform floating-point arithmetic using such a low-precision, high dynamic range number format. In knowingly

66

adopting Dr. Bates' patented computer architectures, Google reaps the very same benefits that

were predicted by Dr. Bates in his patent application more than 10 years ago. As published by

Google and predicted by Dr. Bates in his patent application:

Choosing bfloat16

Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency while maintaining the ability to train accurate deep learning models, all with minimal switching costs from FP32. The physical size of a hardware multiplier scales with the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16 multipliers are about half the size in silicon of a typical FP16 multiplier, and they are *eight times* smaller than an FP32 multiplier!

> PEs implemented according to certain embodiments of the present invention may be relatively small for PEs that can do arithmetic. This means that there are many PEs per unit of resource (e.g., transistor, area, volume), which in turn means that there is a large amount of arithmetic computational power per unit of resource. This enables larger problems to be solved with a given amount of resource than does traditional computer designs. For instance, a digital embodiment of the present invention built as a large silicon chip fabricated with current state of the art technology might perform tens of thousand of arithmetic operations per cycle, as opposed to hundreds in a conventional GPU or a handful in a conventional multicore CPU. These ratios reflect an architectural advantage of embodiments of the present invention that should persist as fabrication technology continues to improve, even as we reach nanotechnology or other implementations for digital and analog computing.

76.   Due to its monitoring of Singular's patents and applications, Google knew of the

application for the '616 patent. For example, Google's attorneys prepared and filed two petitions

for *Inter Partes* Review ("IPR") of the '616 patent.

77.   As a result of Google's infringement of the '616 patent, Singular has suffered

damages in an amount to be determined at trial.

COUNT III
(Google's Infringement of United States Patent No. 11,169,775)

78.   Paragraphs [1-77] are reincorporated by reference as if fully set forth herein.

79.   Google has directly infringed, and continues to directly infringe, literally and/or

by the doctrine of equivalents, at least claim 7 of the '775 patent by making, using, testing,

selling, offering for sale and/or importing into the United States the Accused TPU Computing

Systems alone or in combination with its existing data servers. An Accused TPU Computing

System include a plurality of Accused TPU Devices. Cloud TPU (an Accused TPU Computing

System), in Google's own words, "powers" at least Google Translate, Photos, Search, Assistant,

and Gmail. As published by Google:

Empowering businesses with Google Cloud AI

Machine learning has produced business and research breakthroughs ranging from network security to medical diagnoses. We built the Tensor Processing Unit (TPU) in order to make it possible for anyone to achieve similar breakthroughs. Cloud TPU is the custom-designed machine learning ASIC that powers Google products like Translate, Photos, Search, Assistant, and Gmail. Here's how you can put the TPU and machine learning to work accelerating your company's success, especially at scale.

80. The Accused TPU Computing System comprises "*A computing system,*

comprising: a host computer; a computing chip comprising: a processing element array

comprising a plurality of first processing elements, wherein the plurality of first processing

elements is no less than 5000 in number, wherein each of a first subset of the plurality of first

processing elements is positioned at a first edge of the processing element array, and wherein

each of a second subset of the plurality of first processing elements is positioned in the interior

of the processing element array; an input-output unit connected to each of the first subset of the

plurality of first processing elements; a plurality of processing element connections, each

processing element connection connecting one of the plurality of first processing elements with

another of the plurality of first processing elements, wherein each of the plurality of first

processing elements is connected to at least one other of the plurality of first processing elements

by at least one of the plurality of processing element connections; a plurality of memory units,

wherein each of the plurality of first processing elements is associated with a corresponding one

of the plurality of memory units, and wherein each of the plurality of memory units is local to its

associated one of the plurality of first processing elements." *See* paragraphs 65 to 71 inclusive.

81.   In the computing chip (an Accused TPU Device, which can be a TPUv4 chip or a

TPUv5 chip) of the Accused TPU Computing System, there is "*a plurality of first arithmetic*

units, wherein each of the plurality of first processing elements has positioned therein at least

one of the plurality of first arithmetic units." As shown below in Figures 1c, 2 and 3 of the

Google '165 patent application, each of the aforementioned first processing elements comprises

one MXU AU ("*first arithmetic unit*"). *See* paragraph 72.
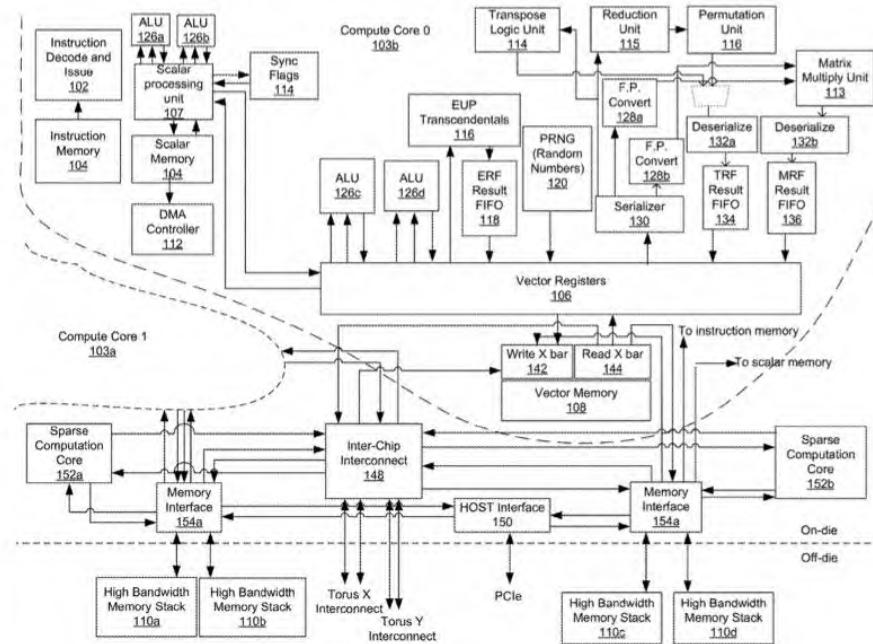
FIG. 1C
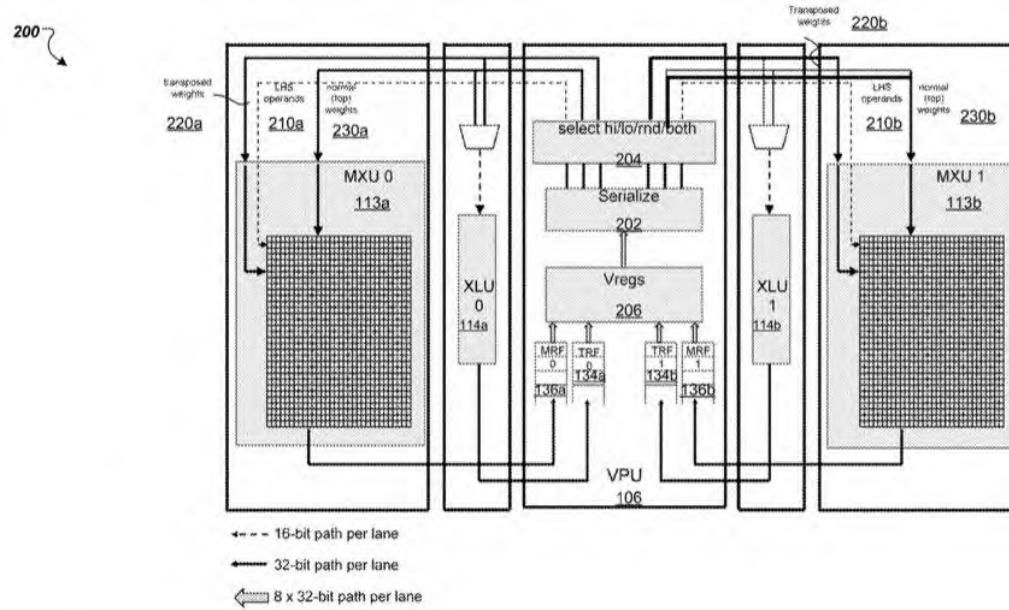
FIG. 2
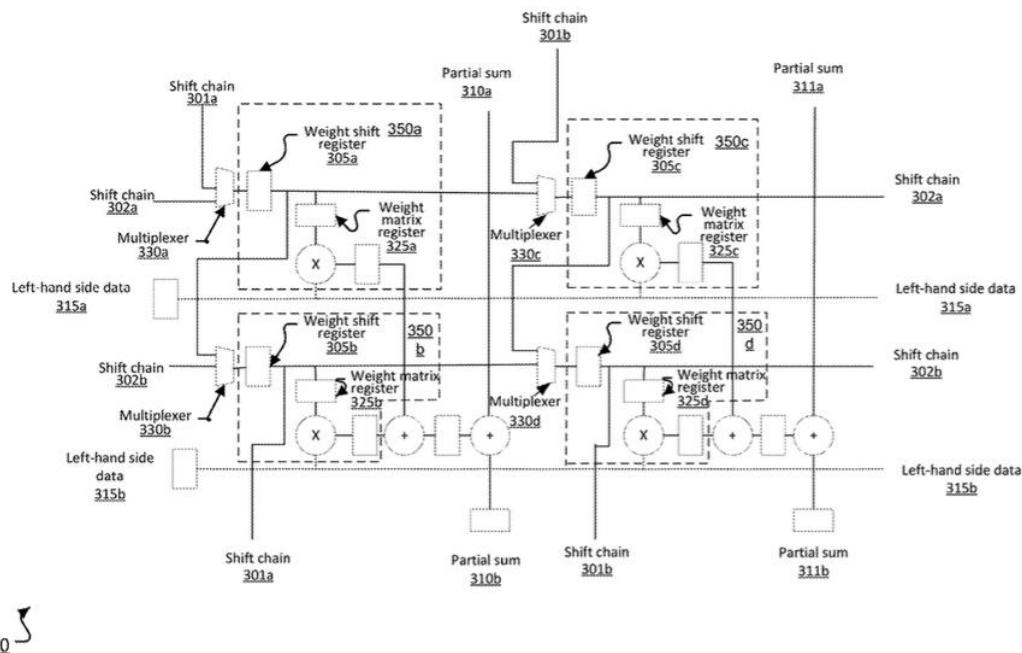
FIG. 3

82.   In the Accused TPU Devices of the Accused TPU Computing System, there is a

"*a plurality of second processing elements; and a plurality of second arithmetic units, wherein*

70

each of the plurality of second processing elements has positioned therein at least one of the

plurality of second arithmetic units." Each TensorCore (as mentioned above, the Accused TPU

Devices each have TensorCores with each TPUv4 having two TensorCores and TPUv5e having

a TensorCore) has a Vector Processing Unit (VPU). Each VPU has 2,048 (16 X 128) processing

elements which each comprise ALUs (arithmetic logic units), which are "*second arithmetic*

units." *See*, *e.g.*, https://codelabs.developers.google.com/codelabs/keras-flowers-data/#2 ("The

VPU handles float32 and int32 computations"). As published by Google:

> Figure 2 below shows a TPU v4 package and four of them mounted on the printed circuit board. Like TPU v3, each TPU v4 contains two *TensorCores* (*TC*). Each TC contains four 128x128 *Matrix Multiply Units* (*MXUs*) and a Vector *Processing Unit* (*VPU*) with 128 lanes (16 ALUs per lane) and a 16 MiB *Vector Memory* (*VMEM*). The two TCs share a 128 MiB Common Memory

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

83.   In the Accused TPU Computing System, there is "*a host connection at least*

partially connecting the input-output unit with the host computer." *See* paragraph 73.

84.   The Accused TPU System, and specifically in the Accused TPU Devices within

the Accused TPU System, the MXU AUs (each a "*first arithmetic unit*") "*each comprises a first*

corresponding multiplier circuit adapted to receive as a first input to the first corresponding

multiplier circuit a first floating point value having a first binary mantissa of width no more than

11 bits and a first binary exponent of width at least 6 bits, and to receive as a second input to the

first corresponding multiplier circuit a second floating point value having a second binary

71

mantissa of width no more than 11 bits and a second binary exponent of width at least 6 bits."

See paragraph 74.

85.   In the Accused TPU Device of the Accused TPU Computing System, "*the first multiplier circuits corresponding to the plurality of first arithmetic units each comprises a first respective plurality of transistors and has no other transistors*." An Accused TPU Device, and each of its constituent parts including the MXU AUs ("*first arithmetic unit*") and the MXU Multiplier Circuit ("*multiplier circuit*" in the MXU AU), each comprise a plurality of transistors. The MXU Multiplier Circuit comprises a plurality of transistors and has no other transistors besides the plurality.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

Feature	TPUv1	TPUv2	TPUv3	TPUv4i	NVIDIA T4
Peak TFLOPS / Chip	92 (8b int)	46 (bf16)	123 (bf16)	138 (bf16/8b int)	65 (ieee fp16)/130 (8b int)
First deployed (GA date)	Q2 2015	Q3 2017	Q4 2018	Q1 2020	Q4 2018
DNN Target	Inference only	Training & Inf.	Training & Inf.	Inference only	Inference only
Network links x Gbits/s / Chip	--	4 x 496	4 x 656	2 x 400	--
Max chips / supercomputer	--	256	1024	--	--
Chip Clock Rate (MHz)	700	700	940	1050	585 / (Turbo 1590)
Idle Power (Watts) Chip	28	53	84	55	36
TDP (Watts) Chip / System	75 / 220	280 / 460	450 / 660	175 / 275	70 / 175
Die Size (mm²)	< 330	< 625	< 700	< 400	545
Transistors (B)	3	9	10	16	14
Chip Technology	28 nm	16 nm	16 nm	7 nm	12 nm
Memory size (on-/off-chip)	28MB / 8GB	32MB / 16GB	32MB / 32GB	144MB / 8GB	18MB / 16GB
Memory GB/s / Chip	34	700	900	614	320 (if ECC is disabled)
MXU Size / Core	1 256x256	1 128x128	2 128x128	4 128x128	8 8x8
Cores / Chip	1	2	2	1	40
Chips / CPUHost	4	4	4	8	8

Table 1. Key characteristics of DSAs. The underlines show changes over the prior TPU generation, from left to right. System TDP includes power for the DSA memory system plus its share of the server host power, e.g., add host TDP/8 for 8 DSAs per host.

86.   In the Accused TPU Device of the Accused TPU Computing System, "*the plurality of second arithmetic units each comprises a second corresponding multiplier circuit adapted to receive as inputs to the second corresponding multiplier circuit two floating point*

values each of width at least 32 bits." As mentioned above, each TensorCore (each TPUv4 has

two TensorCores and TPUv5e has a TensorCore, as explained above) has a Vector Processing

Unit (VPU). As mentioned above, each VPU has 2,048 (16 x 128) processing elements which

each comprise ALUs (arithmetic logic units), which are arithmetic units. Half of these ALUs –

1,024 of them—each comprise a multiplier circuit adapted to receive as inputs two floating point

numbers that are each of a width that is least 32 bits wide. *See*, *e.g.*,

https://codelabs.developers.google.com/codelabs/keras-flowers-data/#2 ("The VPU handles

float32 and int32 computations.") As published by Google:

> Figure 2 below shows a TPU v4 package and four of them
> mounted on the printed circuit board. Like TPU v3, each TPU v4
> contains two *TensorCores* (*TC*). Each TC contains four 128x128
> *Matrix Multiply Units* (*MXUs*) and a Vector *Processing Unit* (*VPU*)
> with 128 lanes (16 ALUs per lane) and a 16 MiB Vector *Memory*
> (*VMEM*). The two TCs share a 128 MiB Common Memory

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

> [0046]   The computational unit includes vector registers,
> i.e., 32 vector registers, in a vector processing unit (106) that
> can be used for both floating point operations and integer
> operations. The computational unit includes two arithmetic
> logic units (ALUs) (126c-d) to perform computations. One
> ALU (126c) performs floating point addition and the other
> ALU (126d) performs floating point multiplication. Both

87.   In the Accused TPU Device of the Accused TPU Computing System, "*the second*

multiplier circuits corresponding to the plurality of second arithmetic units each comprises a

second respective plurality of transistors." An Accused TPU Device, and each of its constituent

parts including the VPU ALUs ("*second arithmetic unit*") and each VPU ALUs multiplier circuit

("*second multiplier circuit*"), each comprise a plurality of transistors.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

Feature	TPUv1	TPUv2	TPUv3	TPUv4i	NVIDIA T4
Peak TFLOPS / Chip	92 (8b int)	46 (bf16)	123 (bf16)	138 (bf16/8b int)	65 (ieee fp16)/130 (8b int)
First deployed (GA date)	Q2 2015	Q3 2017	Q4 2018	Q1 2020	Q4 2018
DNN Target	Inference only	Training & Inf.	Training & Inf.	Inference only	Inference only
Network links x Gbits/s / Chip	--	4 x 496	4 x 656	2 x 400	--
Max chips / supercomputer	--	256	1024	--	--
Chip Clock Rate (MHz)	700	700	940	1050	585 / (Turbo 1590)
Idle Power (Watts) Chip	28	53	84	55	36
TDP (Watts) Chip / System	75 / 220	280 / 460	450 / 660	175 / 275	70 / 175
Die Size (mm²)	< 330	≤ 625	< 700	< 400	545
Transistors (B)	3	9	10	16	14
Chip Technology	28 nm	16 nm	16 nm	7 nm	12 nm
Memory size (on-/off-chip)	28MB / 8GB	32MB / 16GB	32MB / 32GB	144MB / 8GB	18MB / 16GB
Memory GB/s / Chip	34	700	900	614	320 (if ECC is disabled)
MXU Size / Core	1 256x256	1 128x128	2 128x128	4 128x128	8 8x8
Cores / Chip	1	2	2	1	40
Chips / CPUHost	4	4	4	8	8

Table 1. Key characteristics of DSAs. The underlines show changes over the prior TPU generation, from left to right. System TDP includes power for the DSA memory system plus its share of the server host power, e.g., add host TDP/8 for 8 DSAs per host.

88.   In the Accused TPU Devices of the Accused TPU Computing System, "*each of*

the second respective pluralities of transistors of the second multiplier circuits corresponding to

the plurality of second arithmetic units exceeds in number each of the first respective pluralities

of transistors of the first multiplier circuits corresponding to the plurality of first arithmetic

units." Each MXU Reduced Precision Multiply Cells as defined above is smaller than each

multiplier circuit in the VPU ALU. Google engineer Jeffrey Dean, the head of Google Brain,

expressly admitted this:

> Furthermore, one major area & power cost of multiplier circuits for a
> floating point format with M mantissa bits is the $(M+1) \times (M+1)$ array of
> full adders (that are needed for multiplying together the mantissa portions

of the two input numbers. The IEEE fp32, IEEE fp16 and bfloat16 formats need 576 full adders, 121 full adders, and 64 full adders, respectively. ***Because multipliers for the bfloat16 format require so much less circuitry***, it is possible to put more multipliers in the same chip area and power budget, thereby meaning that ML accelerators employing this format can have higher flops/sec and flops/Watt, all other things being equal.

Dean, Jeffrey. (2020). 1.1 *The Deep Learning Revolution and Its Implications for Computer Architecture and Chip Design*. 8-14. 10.1109/ISSCC19947.2020.9063049 (emphasis added).

This fact was further confirmed in a paper published by the team of Google engineers responsible for designing and building the accused TPUs (including, *inter alia*, Norman Jouppi and David Patterson):

	Operation	Picojoules per Operation		
		45 nm	7 nm	45 / 7
+	Int 8	0.03	0.007	4.3
	Int 32	0.1	0.03	3.3
	BFloat 16	--	0.11	--
	IEEE FP 16	0.4	0.16	2.5
	IEEE FP 32	0.9	0.38	2.4
×	Int 8	0.2	0.07	2.9
	Int 32	3.1	1.48	2.1
	BFloat 16	--	0.21	--
	IEEE FP 16	1.1	0.34	3.2
	IEEE FP 32	3.7	1.31	2.8
SRAM	8 KB SRAM	10	7.5	1.3
	32 KB SRAM	20	8.5	2.4
	1 MB SRAM[1]	100	14	7.1
GeoMean[1]		--	--	2.6
		Circa 45 nm	Circa 7 nm	
DRAM	DDR3/4	1300[2]	1300[2]	1.0
	HBM2	--	250-450[2]	--
	GDDR6	--	350-480[2]	--

Table 2. Energy per Operation: 45 nm [16] vs 7 nm. Memory is pJ per 64-bit access.

Jouppi, Norman, *et al.*. "Ten Lessons From Three Generations Shaped Google's TPUv4i : Industrial Product," in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, Valencia, Spain, 2021 pp. 1-14 at 3 (emphasis added). According to the above table, a BFloat 16 multiplier requires less than 20% as much energy per operation as an IEEE FP32 multiplier (both made using the same 7 nm semiconductor fabrication process). The

lower power requirements of BFloat16 multipliers is a result of the fact that they include fewer

transistors than full-precision IEEE FP32 multipliers

89.   In knowingly adopting Dr. Bates' patented computer architectures, Google reaps

the very same benefits that were predicted by Dr. Bates in his patent application more than 10

years ago. As published by Google and predicted by Dr. Bates in his patent application:

Choosing bfloat16

Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency
while maintaining the ability to train accurate deep learning models, all with minimal
switching costs from FP32. The physical size of a hardware multiplier scales with
the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16
multipliers are about half the size in silicon of a typical FP16 multiplier, and they are
eight times smaller than an FP32 multiplier!

> PEs implemented according to certain embodiments of the
> present invention may be relatively small for PEs that can do
> arithmetic. This means that there are many PEs per unit of
> resource (e.g., transistor, area, volume), which in turn means
> that there is a large amount of arithmetic computational
> power per unit of resource. This enables larger problems to be
> solved with a given amount of resource than does traditional
> computer designs. For instance, a digital embodiment of the
> present invention built as a large silicon chip fabricated with
> current state of the art technology might perform tens of
> thousand of arithmetic operations per cycle, as opposed to
> hundreds in a conventional GPU or a handful in a conven-
> tional multicore CPU. These ratios reflect an architectural
> advantage of embodiments of the present invention that
> should persist as fabrication technology continues to
> improve, even as we reach nanotechnology or other imple-
> mentations for digital and analog computing.

90.   Due to its monitoring of Singular's patents and applications, Google knew of the

application for the '775 patent prior to the issuance of the patent on November 9, 2021. For

example, Google's attorneys prepared and filed two IPR petitions for *Inter Partes* Review

("IPR") of patents related to the '775 patent.

91.   As a result of Google's infringement of the '775 patent, Singular has suffered

damages in an amount to be determined at trial.

COUNT IV
(Google's Infringement of United States Patent No. 11,169,659)

92.   Paragraphs [1-91] are reincorporated by reference as if fully set forth herein.

93.   Google has directly infringed, and continues to directly infringe, literally and/or by the doctrine of equivalents, at least claim 1 of the '659 patent by making, using, testing, selling, offering for sale and/or importing into the United States the Accused TPU Devices. The Accused TPU Devices, in Google's own words, "power" at least Google Translate, Photos, Search, Assistant, and Gmail, as published by Google:

Empowering businesses with Google Cloud AI

Machine learning has produced business and research breakthroughs ranging from network security to medical diagnoses. We built the Tensor Processing Unit (TPU) in order to make it possible for anyone to achieve similar breakthroughs. Cloud TPU is the custom-designed machine learning ASIC that powers Google products like Translate, Photos, Search, Assistant, and Gmail. Here's how you can put the TPU and machine learning to work accelerating your company's success, especially at scale.

94.   The Accused TPU Devices share a similar architecture in that they all perform low precision high dynamic range arithmetic operations, specifically multiplication of two traditional high precision floating point value at reduced bfloat16 precision.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The following table shows the key specifications for a v4 TPU Pod.

Key specifications	v4 Pod values
Peak compute per chip	275 teraflops (bf16 or int8)
HBM2 capacity and bandwidth	32 GiB, 1200 GBps
Measured min/mean/max power	90/170/192 W
TPU Pod size	4096 chips
Interconnect topology	3D mesh
Peak compute per Pod	1.1 exaflops (bf16 or int8)
All-reduce bandwidth per Pod	1.1 PB/s
Bisection bandwidth per Pod	24 TB/s

The following diagram illustrates a TPU v4 chip.

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

The following diagram illustrates a TPU v5e chip.

The following table shows the key chip specifications and their values for v5e.

Key chip specifications	v5e values
Peak compute per chip (bf16)	197 TFLOPs
Peak compute per chip (Int8)	393 TFLOPs
HBM2 capacity and bandwidth	16 GB, 819 GBps
Interchip Interconnect BW	1600 Gbps

95.   Each of the Accused TPU Devices, which can be a TPUv4 chip or a TPUv5 chip, is an example of a "*silicon chip that has a clock*," as claimed by the '714 patent. As published by Google:

TPU chip

A TPU chip contains one or more TensorCores. The number of TensorCores depend on the version of the TPU chip. Each TensorCore consists of one or more matrix-multiply units (MXUs), a vector unit, and a scalar unit.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

Feature	TPUv1	TPUv2	TPUv3	TPUv4i	NVIDIA T4
Peak TFLOPS / Chip	92 (8b int)	46 (bf16)	123 (bf16)	138 (bf16/8b int)	65 (ieee fp16)/130 (8b int)
First deployed (GA date)	Q2 2015	Q3 2017	Q4 2018	Q1 2020	Q4 2018
DNN Target	Inference only	Training & Inf.	Training & Inf.	Inference only	Inference only
Network links x Gbits/s / Chip	--	4 x 496	4 x 656	2 x 400	--
Max chips / supercomputer	--	256	1024	--	--
Chip Clock Rate (MHz)	700	700	940	1050	585 / (Turbo 1590)
Idle Power (Watts) Chip	28	53	84	55	36
TDP (Watts) Chip / System	75 / 220	280 / 460	450 / 660	175 / 275	70 / 175
Die Size (mm^2)	< 330	≤ 625	< 700	< 400	545
Transistors (B)	3	9	10	16	14
Chip Technology	28 nm	16 nm	16 nm	7 nm	12 nm
Memory size (on-/off-chip)	28MB / 8GB	32MB / 16GB	32MB / 32GB	144MB / 8GB	18MB / 16GB
Memory GB/s / Chip	34	700	900	614	320 (if ECC is disabled)
MXU Size / Core	1 256x256	1 128x128	2 128x128	4 128x128	8 8x8
Cores / Chip	1	2	2	1	40
Chips / CPUHost	4	4	4	8	8

Table 1. Key characteristics of DSAs. The underlines show changes over the prior TPU generation, from left to right. System TDP includes power for the DSA memory system plus its share of the server host power, e.g., add host TDP/8 for 8 DSAs per host.

96.   A method performed by the each Accused TPU Device comprises "*completing, in a single cycle of the clock, using the silicon chip, at least tens of thousands of first multiplication operations*." Each Accused TPU Device (which can be a TPUv4 chip or a TPUv5 chip), comprises one or more MXUs, each of which contains a systolic array having 128 x 128 "multiply-accumulators." Each of these multiply-accumulators includes a multiplier circuit (MXU Multiplier Circuit). *See* paragraph 52. The MXU Multiplier Circuits each perform multiplication operations as described above. Collectively, the MXU Multiplier Circuits perform at least tens of thousands of first multiplication operations per clock cycle (138 bfloat16 TFLOPS with a clock rate of 1050MHz, which means the TPUv4 chip is performing ≈ 131,000 bfloat16 multiplication operations per clock cycle, and since TPUv5 chip has half the MXUs as a TPUv4 chip, TPUv5 chips perform ≈ 65,000 bfloat16 multiplication operations per clock cycle).

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

Feature	TPUv1	TPUv2	TPUv3	TPUv4i	NVIDIA T4
Peak TFLOPS / Chip	92 (8b int)	46 (bf16)	123 (bf16)	138 (bf16/8b int)	65 (ieee fp16)/130 (8b int)
First deployed (GA date)	Q2 2015	Q3 2017	Q4 2018	Q1 2020	Q4 2018
DNN Target	Inference only	Training & Inf.	Training & Inf.	Inference only	Inference only
Network links x Gbits/s / Chip	--	4 x 496	4 x 656	2 x 400	--
Max chips / supercomputer	--	256	1024	--	--
Chip Clock Rate (MHz)	700	700	940	1050	585 / (Turbo 1590)
Idle Power (Watts) Chip	28	53	84	55	36
TDP (Watts) Chip / System	75 / 220	280 / 460	450 / 660	175 / 275	70 / 175
Die Size (mm²)	< 330	≤ 625	< 700	< 400	545
Transistors (B)	3	9	10	16	14
Chip Technology	28 nm	16 nm	16 nm	7 nm	12 nm
Memory size (on-/off-chip)	28MB / 8GB	32MB / 16GB	32MB / 32GB	144MB / 8GB	18MB / 16GB
Memory GB/s / Chip	34	700	900	614	320 (if ECC is disabled)
MXU Size / Core	1 256x256	1 128x128	2 128x128	4 128x128	8 8x8
Cores / Chip	1	2	2	1	40
Chips / CPUHost	4	4	4	8	8

Table 1. Key characteristics of DSAs. The underlines show changes over the prior TPU generation, from left to right. System TDP includes power for the DSA memory system plus its share of the server host power, e.g., add host TDP/8 for 8 DSAs per host.

97.   In the Accused TPU Device, "*each of the first multiplication operations operates on a respective first numerical input value represented using a first floating point representation that has a signed binary mantissa of no more than 11 bits and a signed binary exponent of at least 6 bits, and a respective second numerical input value represented using a second floating point representation.*"

a.   The Accused TPU Devices each contain TensorCores which each have one of more MXUs. Specifically with respect to the Accused TPU Devices, each TPUv4 chip has 8 MXUs (four MXUs per TensorCore, 2 TensorCores per chip), and each TPUv5 chip has 4 MXUs (four MXUs per TensorCore, 1 TensorCore per chip). As published by Google:

TPU chip

A TPU chip contains one or more TensorCores. The number of TensorCores depend on the version of the TPU chip. Each TensorCore consists of one or more matrix-multiply units (MXUs), a vector unit, and a scalar unit.

An MXU is composed of 128 x 128 multiply-accumulators in a systolic array. MXUs provide the bulk of the compute power in a TensorCore. Each MXU is capable of performing 16K multiply-accumulate operations per cycle. All multiplies take bfloat16 inputs, but all accumulations are performed in FP32 number format.

The vector unit is used for general computation such as activations and softmax. The scalar unit is used for control flow, calculating memory addresses, and other maintenance operations.

TensorCores

TPU chips have one or two TensorCores to run matrix multiplication. Similar to v2 and v3 Pods, v5e has one TensorCore per chip. By contrast, v4 Pods have 2 TensorCores per chip. For more information about TensorCores, see ACM article.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The following table shows the key specifications for a v4 TPU Pod.

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

b.   Each MXU contains a systolic array having 128 x 128 "multiply-accumulators," each of which includes a multiplier circuit (MXU Multiplier Circuit). The MXU Multiplier Circuit is adapted to complete *"a first multiplication operation."* As published by Google:

> An MXU is composed of 128 x 128 multiply-accumulators in a systolic array. MXUs provide the bulk of the compute power in a TensorCore. Each MXU is capable of performing 16K multiply-accumulate operations per cycle. All multiplies take bfloat16 inputs, but all accumulations are performed in FP32 number format.

> The primary task for TPUs is matrix processing, which is a combination of multiply and accumulate operations. TPUs contain thousands of multiply-accumulators that are directly connected to each other to form a large physical matrix. This is called a systolic array architecture. Cloud TPU v3, contain two systolic arrays of 128 x 128 ALUs, on a single processor.

82

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

c.   That is, a MXU Multiplier Circuit is adapted to perform a multiplication (i.e.,

arithmetic) operation on two input bfloat16 numerical values so that the

multiplication operation is carried out in "bfloat16 format." Such an operation

(e.g., "X[2,0]*W[0,0]" in the example equation for Y[2,0] that Google provides

below) is performed in "bfloat16 format" by the MXU as a whole.:

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Systolic array

The MXU implements matrix multiplications in hardware using a so-called "systolic array" architecture in which data elements flow through an array of hardware computation units. (In medicine, 'systolic' refers to heart contractions and blood flow, here to the flow of data.)

The basic element of a matrix multiplication is a dot product between a line from one matrix and a column from the other matrix (see illustration at the top of this section). For a matrix multiplication Y=X*W, one element of the result would be:

```
Y[2,0] = X[2,0]*W[0,0] + X[2,1]*W[1,0] + X[2,2]*W[2,0] + ... + X[2,n]*W[n,0]
```


Illustration: a dense neural network layer as a matrix multiplication, with a batch of eight images processed through the neural network at once. Please run through one line x column multiplication to verify that it is indeed doing a weighted sum of all the pixels values of an image. Convolutional layers can be represented as matrix multiplications.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

System Architecture 🔖 ▾   Send feedback

Tensor Processing Units (TPUs) are application specific integrated circuits (ASICs) designed by Google to accelerate machine learning workloads. Cloud TPU is a Google Cloud service that makes TPUs available as a scalable resource.

f.   The "*first numerical input value*" for each individual MXU Multiplier Circuit is the signal representing a bfloat16 value that is multiplied by the MXU Multiplier Circuit. As published by Google:

> Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

> On a GPU, one would program this dot product into a GPU "core" and then execute it on as many "cores" as are available in parallel to try and compute every value of the resulting matrix at once. If the resulting matrix is 128x128 large, that would require 128x128=16K "cores" to be available which is typically not possible. The largest GPUs have around 4000 cores. A TPU on the other hand uses the bare minimum of hardware for the compute units in the MXU: just `bfloat16 x bfloat16 => float32` multiply-accumulators, nothing else. These are so small that a TPU can implement 16K of them in a 128x128 MXU and process this matrix multiplication in one go.

Illustration: the MXU systolic array. The compute elements are multiply-accumulators. The values of one matrix are loaded into the array (red dots). Values of the other matrix flow through the array (grey dots). Vertical lines propagate the values up. Horizontal lines propagate partial sums. It is left as an exercise to the user to verify that as the data flows through the array, you get the result of the matrix multiplication coming out of the right side.

85

Let's see how a systolic array executes the neural network calculations. At first, the TPU loads the parameters from memory into the matrix of multipliers and adders.

Then, the TPU loads data from memory. As each multiplication is executed, the result will be passed to the next multipliers while taking the summation at the same time. So the output will be the summation of all multiplication results between data and parameters. During the whole process of massive calculations and data passing, no memory access is required at all.

g.   The "*first numerical input value*" is a bloat16 number. Bfloat16 numbers are

floating point representations of "*numerical values*." As published by Google:

> Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU),   a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively,   these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

86

Let's see how a systolic array executes the neural network calculations. At first, the TPU loads the parameters from memory into the matrix of multipliers and adders.

Then, the TPU loads data from memory. As each multiplication is executed, the result will be passed to the next multipliers while taking the summation at the same time. So the output will be the summation of all multiplication results between data and parameters. During the whole process of massive calculations and data passing, no memory access is required at all.

Single-precision floating-point format

From Wikipedia, the free encyclopedia

Single-precision floating-point format is a computer number format, usually occupying 32 bits in computer memory; it represents a wide dynamic range of numeric values by using a floating radix point.

A floating-point variable can represent a wider range of numbers than a fixed-point variable of the same bit width at

h.   Bfloat16 numbers are floating point representations that have a signed binary exponent of 8 bits, and a signed binary mantissa of 8 bits, which means the Accused TPU Device multiplication operations operate on a first numerical input value represented using a signed binary mantissa of no more than 11 bits and a signed binary exponent of at least 6 bits. Because each MXU Multiplier Circuit takes two such inputs (i.e., in bfloat16 format), each MXU Multiplier Circuit multiplication operation also operates on a "*second numerical input values represented using a second floating point representation*."

87

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Choosing bfloat16

Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency while maintaining the ability to train accurate deep learning models, all with minimal switching costs from FP32. The physical size of a hardware multiplier scales with the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16 multipliers are about half the size in silicon of a typical FP16 multiplier, and they are *eight times* smaller than an FP32 multiplier!

(c) bfloat16: Brain Floating Point Format   Range: ~$1e^{-38}$ to ~$3e^{38}$

Exponent: 8 bits   Mantissa (Significand): 7 bits

| S | E | E | E | E | E | E | E | E | M | M | M | M | M | M | M |

98.   In the multiplication operations completed by the Accused TPU Device, "*the number of the first multiplication operations is at least 1000 more than three times the maximum number of traditional high-precision multiplication operations on floating point numbers at least 32 bits wide that the silicon chip is adapted to complete in a single cycle of the clock.*" A TPUv4 chip has eight MXUs (two TensorCores per TPUv4, and four MXUs per TensorCore), while a TPUv5 chip has four MXUs (one TensorCores per TPUv5e, and four MXUs per TensorCore). Each MXU has 16,384 MXU Multiplier Circuits as shown above. Therefore, a TPUv4 chip can complete 131,072 such "*first multiplication operations*" (in a single cycle of the clock) and a TPUv5 chip can complete 65,536 such "*first multiplication operations*" (in a single cycle of the

88

clock). Each TensorCore (each TPUv4 has two TensorCores and TPUv5e has a TensorCore, as explained above) also has a Vector Processing Unit (VPU). Each VPU has 2,048 (16 x 128) ALUs, half of which are custom silicon arithmetic elements in the silicon chip (Accused TPU Device) adapted to perform on each clock cycle the operations of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide. *See*, *e.g.*, https://codelabs.developers.google.com/codelabs/keras-flowers-data/#2 ("The VPU handles float32 and int32 computations.") Therefore, each TPUv4 chip can complete 131,072 of the "*first multiplication operations (in a single cycle of the clock)*" and 2,048 of the "*traditional high-precision multiplication operations on floating point numbers at least 32 bits wide… in a single cycle of the clock*." Each TPUv4 chip can complete 65,536 of the "*first multiplication operations (in a single cycle of the clock)*" and 1,024 of the "*traditional high-precision multiplication operations on floating point numbers at least 32 bits wide… in a single cycle of the clock*". Since 131,072 is at least 1000 more than three times 2,048, and since 65,536 is at least 1000 more than three times 1,024, in the Accused TPU Devices, "*the number of the first multiplication operations is at least 1000 more than three times the maximum number of traditional high-precision multiplication operations on floating point numbers at least 32 bits wide that the silicon chip is adapted to complete in a single cycle of the clock.*" As published by Google:

System Architecture 🔖 ▾   Send feedback

Tensor Processing Units (TPUs) are application specific integrated circuits (ASICs) designed by Google to accelerate machine learning workloads. Cloud TPU is a Google Cloud service that makes TPUs available as a scalable resource.

TPUs are designed to perform matrix operations quickly making them ideal for machine learning workloads. You can run machine learning workloads on TPUs using frameworks such as TensorFlow, Pytorch, and JAX.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU
System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

Figure 2 below shows a TPU v4 package and four of them mounted on the printed circuit board. Like TPU v3, each TPU v4 contains two *TensorCores* (*TC*). Each TC contains four 128x128 *Matrix Multiply Units* (*MXUs*) and a *Vector* Processing Unit (*VPU*) with 128 lanes (16 ALUs per lane) and a 16 MiB *Vector* Memory (*VMEM*). The two TCs share a 128 MiB Common Memory

[0046] The computational unit includes vector registers, i.e., **32** vector registers, in a vector processing unit (**106**) that can be used for both floating point operations and integer operations. The computational unit includes two arithmetic logic units (ALUs) (**126***c-d*) to perform computations. One ALU (**126***c*) performs floating point addition and the other ALU (**126***d*) performs floating point multiplication. Both

99.   Collectively, the MXU Multiplier Circuits perform at least tens of thousands of

the "*first multiplication operations*" per clock cycle (138 bfloat16 TFLOPS with a clock rate of

1050MHz, which means the TPUv4 chip is performing ≈ 131,000 bfloat16 multiplication

operations per clock cycle, and since TPUv5 chip has half the MXUs as a TPUv4 chip, the

TPUv5 chip is performing ≈ 65,000 bfloat16 multiplication operations per clock cycle). By

contrast, TPUv4 can only complete 2,048, and TPUv5e can only complete 1,024, of the

"*traditional high-precision multiplication operations on floating point numbers at least 32 bits*

wide… in a single cycle of the clock."

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

Feature	TPUv1	TPUv2	TPUv3	TPUv4i	NVIDIA T4
Peak TFLOPS / Chip	92 (8b int)	46 (bf16)	123 (bf16)	138 (bf16/8b int)	65 (ieee fp16)/130 (8b int)
First deployed (GA date)	Q2 2015	Q3 2017	Q4 2018	Q1 2020	Q4 2018
DNN Target	Inference only	Training & Inf.	Training & Inf.	Inference only	Inference only
Network links x Gbits/s / Chip	--	4 x 496	4 x 656	2 x 400	--
Max chips / supercomputer	--	256	1024	--	--
Chip Clock Rate (MHz)	700	700	940	1050	585 / (Turbo 1590)
Idle Power (Watts) Chip	28	53	84	55	36
TDP (Watts) Chip / System	75 / 220	280 / 460	450 / 660	175 / 275	70 / 175
Die Size (mm²)	< 330	< 625	< 700	< 400	545
Transistors (B)	3	9	10	16	14
Chip Technology	28 nm	16 nm	16 nm	7 nm	12 nm
Memory size (on-/off-chip)	28MB / 8GB	32MB / 16GB	32MB / 32GB	144MB / 8GB	18MB / 16GB
Memory GB/s / Chip	34	700	900	614	320 (if ECC is disabled)
MXU Size / Core	1 256x256	1 128x128	2 128x128	4 128x128	8 8x8
Cores / Chip	1	2	2	1	40
Chips / CPUHost	4	4	4	8	8

Table 1. Key characteristics of DSAs. The underlines show changes over the prior TPU generation, from left to right. System TDP includes power for the DSA memory system plus its share of the server host power, e.g., add host TDP/8 for 8 DSAs per host.

100.   In knowingly adopting Dr. Bates' patented computer architectures, Google reaps

the very same benefits that were predicted by Dr. Bates in his patent application more than 10

years ago. As published by Google and predicted by Dr. Bates in his patent application:

Choosing bfloat16

Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency while maintaining the ability to train accurate deep learning models, all with minimal switching costs from FP32. The physical size of a hardware multiplier scales with the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16 multipliers are about half the size in silicon of a typical FP16 multiplier, and they are *eight times* smaller than an FP32 multiplier!

> PEs implemented according to certain embodiments of the present invention may be relatively small for PEs that can do arithmetic. This means that there are many PEs per unit of resource (e.g., transistor, area, volume), which in turn means that there is a large amount of arithmetic computational power per unit of resource. This enables larger problems to be solved with a given amount of resource than does traditional computer designs. For instance, a digital embodiment of the present invention built as a large silicon chip fabricated with current state of the art technology might perform tens of thousand of arithmetic operations per cycle, as opposed to hundreds in a conventional GPU or a handful in a conventional multicore CPU. These ratios reflect an architectural advantage of embodiments of the present invention that should persist as fabrication technology continues to improve, even as we reach nanotechnology or other implementations for digital and analog computing.

101.   Due to its monitoring of Singular's patents and applications, Google knew of the application for the '659 patent prior to the issuance of the patent on September 26, 2023. For example, Google's attorneys prepared and filed two petitions for *Inter Partes* Review ("IPR") of the 616 patent, patents related to the '659 patent. In each of those petitions, Google identified numerous patents and applications related to the 616 patent, including application serial number US17/029,780, which led to the '659 patent. Thus, at least since 12/22/2022 when Google identified the application in, *inter alia*, its Petition for *Inter Partes* Review in IPR2023-00395, Google has knowledge of the '780 application. Before making such identification, counsel for Google reviewed application serial number US17/029,780.

102.   As a result of Google's infringement of the '659 patent, Singular has suffered damages in an amount to be determined at trial.

COUNT V

(Google's Infringement of United States Patent No. 11,768,660)

103.   Paragraphs [1-102] are reincorporated by reference as if fully set forth herein.

104.   Google has directly infringed, and continues to directly infringe, literally and/or by the doctrine of equivalents, at least claim 1 of the '660 patent by making, using, testing, selling, offering for sale and/or importing into the United States the Accused TPU Devices. The

Accused TPU Devices, in Google's own words, "power" at least Google Translate, Photos,

Search, Assistant, and Gmail, as published by Google:

Empowering businesses with Google Cloud AI

Machine learning has produced business and research breakthroughs ranging from network security to medical diagnoses. We built the Tensor Processing Unit (TPU) in order to make it possible for anyone to achieve similar breakthroughs. Cloud TPU is the custom-designed machine learning ASIC that powers Google products like Translate, Photos, Search, Assistant, and Gmail. Here's how you can put the TPU and machine learning to work accelerating your company's success, especially at scale.

105.   The Accused TPU Devices share a similar architecture in that they all implement

low precision high dynamic range arithmetic operations, specifically multiplication of two

traditional high precision floating point value at reduced bfloat16 precision.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The following table shows the key specifications for a v4 TPU Pod.

Key specifications	v4 Pod values
Peak compute per chip	275 teraflops (bf16 or int8)
HBM2 capacity and bandwidth	32 GiB, 1200 GBps
Measured min/mean/max power	90/170/192 W
TPU Pod size	4096 chips
Interconnect topology	3D mesh
Peak compute per Pod	1.1 exaflops (bf16 or int8)
All-reduce bandwidth per Pod	1.1 PB/s
Bisection bandwidth per Pod	24 TB/s

The following diagram illustrates a TPU v4 chip.

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

The following diagram illustrates a TPU v5e chip.

The following table shows the key chip specifications and their values for v5e.

Key chip specifications	v5e values
Peak compute per chip (bf16)	197 TFLOPs
Peak compute per chip (Int8)	393 TFLOPs
HBM2 capacity and bandwidth	16 GB, 819 GBps
Interchip Interconnect BW	1600 Gbps

106.   An Accused TPU Pod, that groups one or more Accused TPU Devices (one or more TPUv4 chips or one or more TPUv5 chips) therewithin, is an example of a "*device*." As published by Google:

System Architecture 🔖 ▾

Send feedback

Tensor Processing Units (TPUs) are application specific integrated circuits (ASICs) designed by Google to accelerate machine learning workloads. Cloud TPU is a Google Cloud service that makes TPUs available as a scalable resource.

TPUs are designed to perform matrix operations quickly making them ideal for machine learning workloads. You can run machine learning workloads on TPUs using frameworks such as TensorFlow, Pytorch, and JAX.

TPU Pod

A TPU <mark>Pod</mark> is a contiguous set of TPUs grouped together over a specialized network. The number of TPU chips in a TPU <mark>Pod</mark> is dependent on the TPU version.

107.   Each Accused TPU Device has "*a silicon chip comprising a plurality of execution units.*"

a.   The Accused TPU Devices each contain TensorCores which each have one of more MXUs. Specifically with respect to the Accused TPU Devices, each TPUv4 chip has 8 MXUs (four MXUs per TensorCore, 2 TensorCores per chip), and each TPUv5 chip has 4 MXUs (four MXUs per TensorCore, 1 TensorCore per chip). As published by Google:

TPU chip

A TPU chip contains one or more TensorCores. The number of TensorCores depend on the version of the TPU chip. Each TensorCore consists of one or more matrix-multiply units (MXUs), a vector unit, and a scalar unit.

An MXU is composed of 128 x 128 multiply-accumulators in a systolic array. MXUs provide the bulk of the compute power in a TensorCore. Each MXU is capable of performing 16K multiply-accumulate operations per cycle. All multiplies take bfloat16 inputs, but all accumulations are performed in FP32 number format.

The vector unit is used for general computation such as activations and softmax. The scalar unit is used for control flow, calculating memory addresses, and other maintenance operations.

TensorCores

TPU chips have one or two TensorCores to run matrix multiplication. Similar to v2 and v3 Pods, v5e has one TensorCore per chip. By contrast, v4 Pods have 2 TensorCores per chip. For more information about TensorCores, see ACM article.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The following table shows the key specifications for a v4 TPU Pod.

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

      b.   Each MXU contains a systolic array having 128 x 128 "multiply-accumulators,"

which each include a multiplier circuit (MXU Multiplier Circuit). As published by

Google:

> An MXU is composed of 128 x 128 multiply-accumulators in a systolic array. MXUs provide the bulk of the compute power in a TensorCore. Each MXU is capable of performing 16K multiply-accumulate operations per cycle. All multiplies take bfloat16 inputs, but all accumulations are performed in FP32 number format.

> The primary task for TPUs is matrix processing, which is a combination of multiply and accumulate operations. TPUs contain thousands of multiply-accumulators that are directly connected to each other to form a large physical matrix. This is called a systolic array architecture. Cloud TPU v3, contain two systolic arrays of 128 x 128 ALUs, on a single processor.

> Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

> Cloud TPU
> # System Architecture
> Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

      c.   Each of those MXU Multiplier Circuits are paired with circuitry for taking two

32-bit floating point format ("FP32 format" or "float32") values and converting

each to a bfloat16 value (an MXU Multiplier Circuit and said taking/converting

circuitry, collectively an "MXU Reduced Precision Multiply Cell"). An MXU

97

Reduced Precision Multiply Cell is an "*execution unit.*" A plurality of Reduced

Precision Multiply Cells is a plurality of "*execution units*." As published by

Google:

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU),   a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU
System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

d.   In the Accused TPU device, in claim 1 of the '660 patent, each "*execution

unit*," a Reduced Precision Multiply Cell, is paired with a memory circuit as

shown in Figure 3 of the Google '165 patent application. Each such memory unit

is local to its associated processing element. *See also*, *e.g.*,

https://cloud.google.com/tpu/docs/beginners-guide ("the TPU loads the

parameters from memory into the matrix of multipliers and adders.

FIG. 3
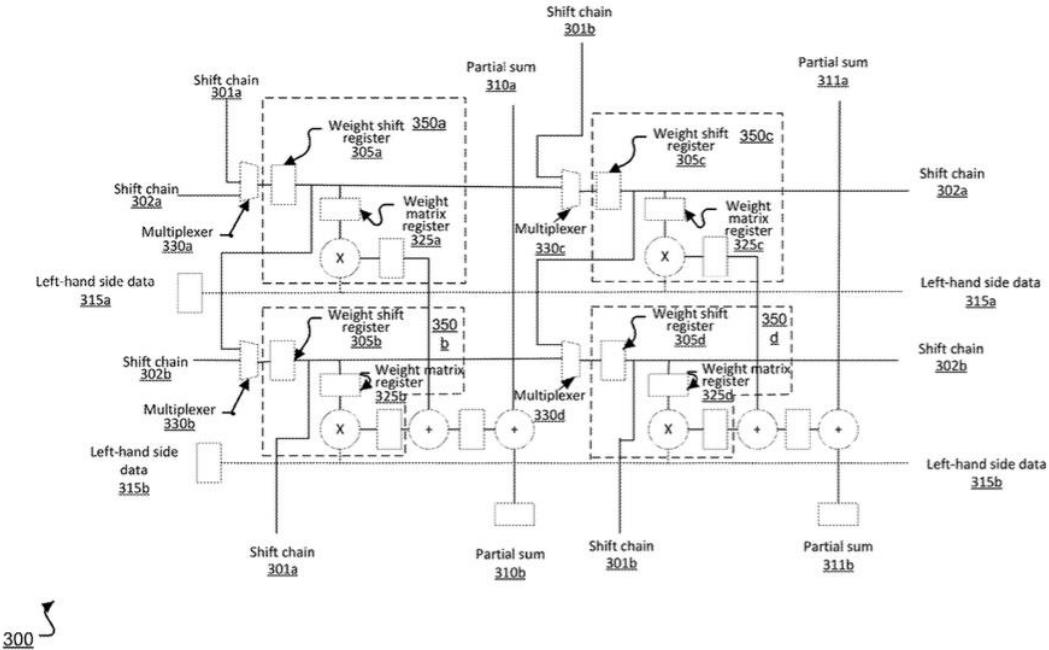
This memory is used, for example, to store "weights" or "parameters" as part of algorithms that relate to neural networks. *See*, *e.g.*, https://www.programmersought.com/article/66614714332/ and previously https://cloud.google.com/tpu/docs/beginners-guide ("the TPU loads the parameters from memory into the matrix of multipliers and adders").

108.   In each Accused TPU Device, "*the plurality of execution units jointly comprise a first plurality of custom silicon arithmetic elements wherein at least one of the first plurality of custom silicon arithmetic elements is adapted to execute a first multiplication operation on one or more first input signals that represent a first numerical value using a floating point representation that has a signed binary mantissa of no more than 11 bits and a signed binary exponent of at least 6 bits, and on one or more second input signals that represent a second numerical value using a floating point representation.*"

99

a.   Each MXU Reduced Precision Multiply Cell (an "*execution unit*") comprises an

MXU Multiplier Circuit paired with circuitry for taking two 32-bit floating point

format ("FP32 format" or "float32") values and converting each to a bfloat16

value. Each of those MXU Multiplier Circuits is a "*custom silicon arithmetic*

element" that is "*adapted to execute a first multiplication operation.*" Each MXU

comprises a systolic array having 128 x 128 "multiply-accumulators," each of

which comprises a MXU Multiplier Circuit. As published by Google:

> An MXU is composed of 128 x 128 multiply-accumulators in a systolic array. MXUs provide the bulk of the compute power in a TensorCore. Each MXU is capable of performing 16K multiply-accumulate operations per cycle. All multiplies take bfloat16 inputs, but all accumulations are performed in FP32 number format.

> The primary task for TPUs is matrix processing, which is a combination of multiply and accumulate operations. TPUs contain thousands of multiply-accumulators that are directly connected to each other to form a large physical matrix. This is called a systolic array architecture. Cloud TPU v3, contain two systolic arrays of 128 x 128 ALUs, on a single processor.

> Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

c.   An MXU Multiplier Circuit is adapted to perform a multiplication (i.e.,

arithmetic) operation on two input signals that were float32 numerical value, but

have been rounded down to bfloat16 numerical values by the time the MXU

Multipier Circuit processes them, so that the multiplication operation is carried

out at "reduced bfloat16 precision." Such an operation (e.g., "X[2,0]*W[0,0]" in

the example equation for Y[2,0] that Google provides below) is a part of a larger

float32 matrix multiplication operation (e.g., Y= X*W in the example Google

provides below) being performed at "reduced bfloat16 precision" by the MXU as

a whole. Since the Accused TPU Device comprises custom silicon, each MXU

Multiplier Circuit is a custom silicon arithmetic element *of the first plurality of*

custom silicon arithmetic elements. As published by Google:

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Systolic array

The MXU implements matrix multiplications in hardware using a so-called "systolic array" architecture in which data elements flow through an array of hardware computation units. (In medicine, "systolic" refers to heart contractions and blood flow, here to the flow of data.)

The basic element of a matrix multiplication is a dot product between a line from one matrix and a column from the other matrix (see illustration at the top of this section). For a matrix multiplication Y=X*W, one element of the result would be:

$$Y[2,0] = X[2,0]*W[0,0] + X[2,1]*W[1,0] + X[2,2]*W[2,0] + ... + X[2,n]*W[n,0]$$

Illustration: a dense neural network layer as a matrix multiplication, with a batch of eight images processed through the neural network at once. Please run through one line x column multiplication to verify that it is indeed doing a weighted sum of all the pixels values of an image. Convolutional layers can be represented as matrix multiplications.

Cloud TPU
System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

System Architecture 🔖 ▾   Send feedback

Tensor Processing Units (TPUs) are application specific integrated circuits (ASICs) designed by Google to accelerate machine learning workloads. Cloud TPU is a Google Cloud service that makes TPUs available as a scalable resource.

    f.   The "*first input signal*" for each individual MXU Multiplier Circuit is the signal representing a bfloat16 value that is multiplied by the MXU Multiplier Circuit. As published by Google:

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

On a GPU, one would program this dot product into a GPU "core" and then execute it on as many "cores" as are available in parallel to try and compute every value of the resulting matrix at once. If the resulting matrix is 128x128 large, that would require 128x128=16K "cores" to be available which is typically not possible. The largest GPUs have around 4000 cores. A TPU on the other hand uses the bare minimum of hardware for the compute units in the MXU: just `bfloat16 x bfloat16 => float32` multiply-accumulators, nothing else. These are so small that a TPU can implement 16K of them in a 128x128 MXU and process this matrix multiplication in one go.

Illustration: the MXU systolic array. The compute elements are multiply-accumulators. The values of one matrix are loaded into the array (red dots). Values of the other matrix flow through the array (grey dots). Vertical lines propagate the values up. Horizontal lines propagate partial sums. It is left as an exercise to the user to verify that as the data flows through the array, you get the result of the matrix multiplication coming out of the right side.

103

Let's see how a systolic array executes the neural network calculations. At first, the TPU loads the parameters from memory into the matrix of multipliers and adders.

Then, the TPU loads data from memory. As each multiplication is executed, the result will be passed to the next multipliers while taking the summation at the same time. So the output will be the summation of all multiplication results between data and parameters. During the whole process of massive calculations and data passing, no memory access is required at all.

g.   The "*first numerical value*" represented by the "*first input signal*," is a bloat16 number. Bfloat16 numbers are floating point representations of "*numerical values*." As published by Google:

> Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU),   a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU
System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

Let's see how a systolic array executes the neural network calculations. At first, the TPU loads the parameters from memory into the matrix of multipliers and adders.

Then, the TPU loads data from memory. As each multiplication is executed, the result will be passed to the next multipliers while taking the summation at the same time. So the output will be the summation of all multiplication results between data and parameters. During the whole process of massive calculations and data passing, no memory access is required at all.

Single-precision floating-point format

From Wikipedia, the free encyclopedia

Single-precision floating-point format is a computer number format, usually occupying 32 bits in computer memory; it represents a wide dynamic range of numeric values by using a floating radix point.

A floating-point variable can represent a wider range of numbers than a fixed-point variable of the same bit width at

h.   The inputs received by each MXU Multiplier Circuit comprise two floating point values having a bfloat16 format. The bfloat16 format used by an MXU Multiplier Circuit has a signed binary exponent bits of 8 bits, and a signed binary mantissa of 8 bits, which means each "*arithmetic element*" of the Accused TPU Device is adapted to execute a multiplication operation on inputs having a signed binary mantissa of width that is no more than 11 bits and a signed binary exponent of width that is at least 6 bits. Because each MXU Multiplier Circuit takes two such inputs (i.e., in bfloat16 format), the MXU Multiplier Circuits also takes "*one or more second input signals that represents a second numerical value using a floating point representation.*"

105

Cloud TPU

System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Choosing bfloat16

Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency while maintaining the ability to train accurate deep learning models, all with minimal switching costs from FP32. The physical size of a hardware multiplier scales with the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16 multipliers are about half the size in silicon of a typical FP16 multiplier, and they are *eight times* smaller than an FP32 multiplier!

(c) bfloat16: Brain Floating Point Format

Range: ~$1e^{-38}$ to ~$3e^{38}$

Exponent: 8 bits	Mantissa (Significand): 7 bits

| S | E | E | E | E | E | E | E | E | M | M | M | M | M | M | M |

109.   In the Accused TPU Device, "*a total number of the first plurality of custom silicon arithmetic elements in the silicon chip that are adapted to execute first multiplication operations exceeds, by at least 1000 more than three times, a total number of second custom silicon arithmetic elements in the silicon chip adapted to perform on each cycle the operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide.*" Each **MXU Multiplier Circuit** is one of the "*custom silicon arithmetic elements. adapted to execute a first multiplication operation on one or more first input signals that represent a first numerical value using a floating point representation that has a signed binary mantissa of no more than 11 bits and a signed binary exponent of at least 6 bits*," as described above. A TPUv4

chip has eight MXUs (two TensorCores per TPUv4, and four MXUs per TensorCore), while a

TPUv5 chip has four MXUs (one TensorCores per TPUv5e, and four MXUs per TensorCore).

Each MXU has 16,384 such "*custom silicon arithmetic elements of the first plurality of custom*

silicon arithmetic elements" (MXU Multiplier Circuits) as shown above. Therefore, a TPUv4

chip has 131,072 such "*custom silicon arithmetic elements*" (MXU Multiplier Circuits) and a

TPUv5 chip has 65,536 such "*custom silicon arithmetic elements*" (MXU Multiplier Circuits).

Each TensorCore (each TPUv4 has two TensorCores and TPUv5e has a TensorCore, as

explained above) also has a Vector Processing Unit (VPU), which like the rest of TPU also

comprises custom silicon. Each VPU has 2,048 (16 x 128) ALUs, half of which are custom

silicon arithmetic elements in the silicon chip (Accused TPU Device) adapted to perform on each

clock cycle the operations of traditional high-precision multiplication on floating point numbers

that are at least 32 bits wide. *See*, *e.g.*, https://codelabs.developers.google.com/codelabs/keras-

flowers-data/#2 ("The VPU handles float32 and int32 computations"). Thus, half of the ALUs in

each VPU are "*second custom silicon arithmetic elements in the silicon chip adapted to perform*

on each cycle the operation of traditional high-precision multiplication on floating point

numbers that are at least 32 bits wide." Therefore, each TPUv4 chip has 131,072 of the "*first*

plurality of custom silicon arithmetic elements" (MXU Multiplier Circuits) and 2,048 (from the

two VPUs within the two TensorCores) "*second custom silicon arithmetic elements*." 131,072

exceeds by at least 1000 more than three times 2,048. Each TPUv5 chip has 65,536 "*custom*

silicon arithmetic elements of the first plurality of custom silicon arithmetic elements"(MXU

Multiplier Circuits) and 1,024 of the "*second custom silicon arithmetic elements*." 65,536

exceeds by at least 1000 more than three times 1,024. Therefore, in the TPU Accused Devices,

there are "*a total number of the first plurality of custom silicon arithmetic elements in the silicon*

chip that are adapted to execute first multiplication operations exceeds, by at least 1000 more

than three times, a total number of second custom silicon arithmetic elements in the silicon chip

adapted to perform on each cycle the operation of traditional high-precision multiplication on

floating point numbers that are at least 32 bits wide." As published by Google:

System Architecture 🔖 ▾   Send feedback

Tensor Processing Units (TPUs) are application specific integrated circuits (ASICs) designed by Google to accelerate machine learning workloads. Cloud TPU is a Google Cloud service that makes TPUs available as a scalable resource.

TPUs are designed to perform matrix operations quickly making them ideal for machine learning workloads. You can run machine learning workloads on TPUs using frameworks such as TensorFlow, Pytorch, and JAX.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

Cloud TPU
System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

Figure 2 below shows a TPU v4 package and four of them mounted on the printed circuit board. Like TPU v3, each TPU v4 contains two *TensorCores* (*TC*). Each TC contains four 128x128 *Matrix Multiply Units* (*MXUs*) and a *Vector* Processing Unit (*VPU*) with 128 lanes (16 ALUs per lane) and a 16 MiB *Vector* Memory (*VMEM*). The two TCs share a 128 MiB Common Memory

[0046] The computational unit includes vector registers, i.e., **32** vector registers, in a vector processing unit (**106**) that can be used for both floating point operations and integer operations. The computational unit includes two arithmetic logic units (ALUs) (**126***c-d*) to perform computations. One ALU (**126***c*) performs floating point addition and the other ALU (**126***d*) performs floating point multiplication. Both

110.   In the Accused TPU Devices, "*the first plurality of custom silicon arithmetic elements are adapted to collectively perform, per cycle, at least tens of thousands of first multiplication operations.*" The "*first plurality of custom silicon arithmetic elements*" are the MXU Multiplier Circuits, as described above. The MXU Multiplier Circuits each perform multiplication operations also as described above. Collectively, the MXU Multiplier Circuits perform at least tens of thousands of first multiplication operations per clock cycle (138 bfloat16 TFLOPS with a clock rate of 1050MHz, which means the TPUv4 chip is performing $\approx$ 131,000 bfloat16 multiplication operations per clock cycle, and since TPUv5 chip has half the MXUs as a TPUv4 Device, the TPUv5 chip is performing $\approx$ 65,000 bfloat16 multiplication operations per clock cycle).

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

Feature	TPUv1	TPUv2	TPUv3	TPUv4i	NVIDIA T4
Peak TFLOPS / Chip	92 (8b int)	46 (bf16)	123 (bf16)	138 (bf16/8b int)	65 (ieee fp16)/130 (8b int)
First deployed (GA date)	Q2 2015	Q3 2017	Q4 2018	Q1 2020	Q4 2018
DNN Target	Inference only	Training & Inf.	Training & Inf.	Inference only	Inference only
Network links x Gbits/s / Chip	--	4 x 496	4 x 656	2 x 400	--
Max chips / supercomputer	--	256	1024	--	--
Chip Clock Rate (MHz)	700	700	940	1050	585 / (Turbo 1590)
Idle Power (Watts) Chip	28	53	84	55	36
TDP (Watts) Chip / System	75 / 220	280 / 460	450 / 660	175 / 275	70 / 175
Die Size (mm^2)	< 330	< 625	< 700	< 400	545
Transistors (B)	3	9	10	16	14
Chip Technology	28 nm	16 nm	16 nm	7 nm	12 nm
Memory size (on-/off-chip)	28MB / 8GB	32MB / 16GB	32MB / 32GB	144MB / 8GB	18MB / 16GB
Memory GB/s / Chip	34	700	900	614	320 (if ECC is disabled)
MXU Size / Core	1 256x256	1 128x128	2 128x128	4 128x128	8 8x8
Cores / Chip	1	2	2	1	40
Chips / CPUHost	4	4	4	8	8

Table 1. Key characteristics of DSAs. The underlines show changes over the prior TPU generation, from left to right. System TDP includes power for the DSA memory system plus its share of the server host power, e.g., add host TDP/8 for 8 DSAs per host.

111.   In knowingly adopting Dr. Bates' patented computer architectures, Google reaps the very same benefits that were predicted by Dr. Bates in his patent application more than 10 years ago. As published by Google and predicted by Dr. Bates in his patent application:

Choosing bfloat16

Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency while maintaining the ability to train accurate deep learning models, all with minimal switching costs from FP32. The physical size of a hardware multiplier scales with the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16 multipliers are about half the size in silicon of a typical FP16 multiplier, and they are *eight times* smaller than an FP32 multiplier!

PEs implemented according to certain embodiments of the present invention may be relatively small for PEs that can do arithmetic. This means that there are many PEs per unit of resource (e.g., transistor, area, volume), which in turn means that there is a large amount of arithmetic computational power per unit of resource. This enables larger problems to be solved with a given amount of resource than does traditional computer designs. For instance, a digital embodiment of the present invention built as a large silicon chip fabricated with current state of the art technology might perform tens of thousand of arithmetic operations per cycle, as opposed to hundreds in a conventional GPU or a handful in a conventional multicore CPU. These ratios reflect an architectural advantage of embodiments of the present invention that should persist as fabrication technology continues to improve, even as we reach nanotechnology or other implementations for digital and analog computing.

112.   As a result of Google's infringement of the '659 patent, Singular has suffered damages in an amount to be determined at trial.

COUNT VI
(Google's Infringement of United States Patent No. 11,842,166)

113.   Paragraphs [1-112] are reincorporated by reference as if fully set forth herein.

114.   Google has directly infringed, and continues to directly infringe, literally and/or by the doctrine of equivalents, at least claim 1 of the '166 patent by making, using, testing, selling, offering for sale and/or importing into the United States the Accused TPU Devices. The Accused TPU Devices, in Google's own words, "power" at least Google Translate, Photos, Search, Assistant, and Gmail, as published by Google:

Empowering businesses with Google Cloud AI

Machine learning has produced business and research breakthroughs ranging from network security to medical diagnoses. We built the Tensor Processing Unit (TPU) in order to make it possible for anyone to achieve similar breakthroughs. Cloud TPU is the custom-designed machine learning ASIC that powers Google products like Translate, Photos, Search, Assistant, and Gmail. Here's how you can put the TPU and machine learning to work accelerating your company's success, especially at scale.

115.   The Accused TPU Devices share a similar architecture in that they all implement low precision high dynamic range arithmetic operations, specifically multiplication of two traditional high precision floating point value at reduced bfloat16 precision.

TPU v4

Each TPU v4 chip contains two TensorCores. Each TensorCore has four MXUs, a vector unit, and a scalar unit. The following table shows the key specifications for a v4 TPU Pod.

Key specifications	v4 Pod values
Peak compute per chip	275 teraflops (bf16 or int8)
HBM2 capacity and bandwidth	32 GiB, 1200 GBps
Measured min/mean/max power	90/170/192 W
TPU Pod size	4096 chips
Interconnect topology	3D mesh
Peak compute per Pod	1.1 exaflops (bf16 or int8)
All-reduce bandwidth per Pod	1.1 PB/s
Bisection bandwidth per Pod	24 TB/s

The following diagram illustrates a TPU v4 chip.

TPU v5e

Each v5e chip contains one TensorCore. Each TensorCore has 4 Matrix Multiply Units (MXU), a vector unit, and a scalar unit.

The following diagram illustrates a TPU v5e chip.

The following table shows the key chip specifications and their values for v5e.

| Key chip specifications	v5e values
Peak compute per chip (bf16)	197 TFLOPs
Peak compute per chip (Int8)	393 TFLOPs
HBM2 capacity and bandwidth	16 GB, 819 GBps
Interchip Interconnect BW	1600 Gbps

116.   Each Accused TPU Device comprises a "*device comprising: at least one instruction memory adapted to store at least one instruction; a silicon chip comprising a plurality of first execution units, wherein each of the plurality of first execution units . . . is adapted to execute a first operation of multiplication: on one or more first input signals that represent a first numerical value using a floating-point representation . . ., and on one or more second input signals that represent a second numerical value using a floating-point representation, to produce one or more first output signals that represent a third numerical value.*" *See* paragraphs 50-52.

117.   In the Accused TPU device, each one of first execution units "*has access to memory local to that execution unit.*" *See* paragraphs 53.

118.   In the Accused TPU Devices, the first numerical values are represented "*using a floating-point representation that has a signed binary mantissa of no more than 11 bits and a signed binary exponent of at least 6 bits*." *See* paragraph 97.

119.   Each Accused TPU Device comprises "*a second execution unit adapted to execute a second operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide; . . . wherein a total number of first execution units in the silicon chip exceeds, by at least 100 more than five times, a total number of execution units in the silicon chip adapted to execute the operation of traditional high-precision multiplication on floating point numbers that are at least 32 bits wide; wherein each of the plurality of first execution units is smaller than the second execution unit*." *See* paragraphs 55-57.

120.   Each Accused TPU Device comprises "*decoding circuitry adapted to decode the at least one instruction received from the at least one instruction memory and to send at least one control signal to at least one of the plurality of first execution units to cause the at least one of the plurality of first execution units to operate according to the at least one instruction*." *See* paragraph 58.

121.   In the Accused TPU Devices, "*the plurality of first execution units are adapted to collectively perform, per cycle, at least tens of thousands of the first operation*." *See* paragraph 59.

122.   As a result of Google's infringement of the '166 patent, Singular has suffered damages in an amount to be determined at trial.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays that the Court:

A.   enter judgment in favor of the Plaintiff on all counts of the Complaint;

B.   award Plaintiff damages as determined at trial;

C.   award Plaintiff treble damages, costs and attorney's fees as a result of

Defendant's willful infringement;

D.   enjoin Defendant's infringement; and

E.   award Plaintiffs such other and further legal and equitable relief as the Court may

deem just and proper.

<u>DEMAND FOR JURY TRIAL</u>

Plaintiff demands a trial by jury on all counts of the complaint.

Dated: January 2, 2024   Respectfully submitted,

                                                    /s/   Kevin Gannon
                                                    Matthew D. Vella (BBO #660171)
                                                    Adam R. Doherty (BBO #669499)
                                                    Kevin Gannon (BBO #640931)
                                                    Brian Seeve (BBO #670455)
                                                    Suhrid Wadekar (BBO #676315)
                                                    Daniel McGonagle (BBO #690084)
                                                    PRINCE LOBEL TYE LLP
                                                    One International Place, Suite 3700
                                                    Boston, MA 02110
                                                    Tel: (617) 456-8000
                                                    Email: mvella@princelobel.com
                                                    Email: adoherty@princelobel.com
                                                    Email: kgannon@princelobel.com
                                                    Email: bseeve@princelobel.com
                                                    Email: swadekar@princelobel.com
                                                    Email: dmcgonagle@princelobel.com

                                                    Kerry L. Timbers (BBO #552293)
                                                    SUNSTEIN LLP
                                                    100 High Street
                                                    Boston, MA 02110
                                                    Tel: (617) 443-9292
                                                    Email: ktimbers@sunsteinlaw.com

                                                    ATTORNEYS FOR THE PLAINTIFF

CERTIFICATE OF SERVICE

I certify that all counsel of record who have consented to electronic service are being served with a copy of this document via the Court's CM/ECF system on January 2, 2024.

/s/   Kevin Gannon